1992 / OKTÓBER

ÁRA: 196 FT

ALAPLAP

MIKROSZÁMÍTÓGÉP MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL

A TopSpeed család

Csodatömb és társai

Források és kútfők

Humorérzék: kötelező!



A HÓNAP TÉMÁJA:

MENTSÜK, AMI MENTHETŐ!

SzabaDOS, LINK magatartás

Öreg szoftver nem rossz szoftver

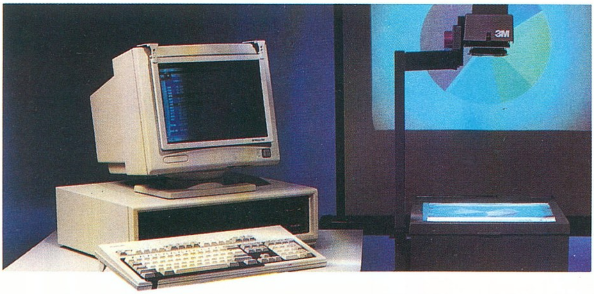
CAMP — kempingszéken

A MÁGNESLEMEZEN:

Antivírus-frissítő
LINKségek — mutatóba
Költők, versek, snobolok
Portok csereberéje
Memóriajáték

Világóra — világtérkép

A látvány meggyőző ereje



Teljes rendszer a képalkotásban

- hordozható és asztali írásvetítők,
- LCD kivetítők, video kivetítők,
- fóliák írásvetítőkhöz, nyomtatókhoz,
- tollak, kiegészítők.

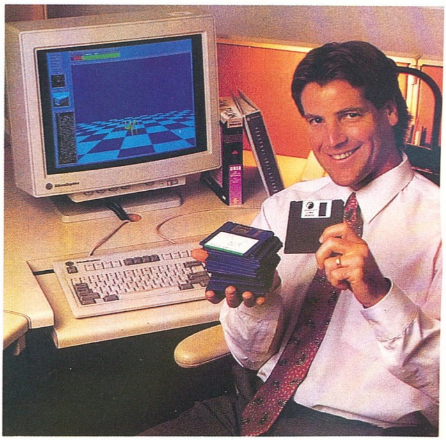
Privacy Filter – speciális monitorszűrők amikor a képernyőn megjelenő információ csak Önre tartozik.

3M **Újdonságokkal is mindig a csúcson**

21 MB 3M Floptical® diskette technológia

Számtalan oka van, hogy Ön ezt válassza

- 21 MB névleges formatált kapacitás,
- a Floptical® diskette technológia hamarosan eléri a 80 MB kapacitást,
- teljes írás/olvasás kompatibilitás a standard 3,5" diskettekkel,
- átlagos elérési idő 65 ms,
- fokozott megbízhatóság
  - rendkívül tartós alapanyag
  - speciális felületi védőbevonat
  - beépített hibaellenőrzés/korrigálás
- ideális PC, Laptop, munkaállomás felhasználói körben is,
- széles felhasználási terület
  - grafikus alkalmazás
  - nagy adatbázisok kezelése
  - multimédia előadások
  - backup adatbiztosítás
  - adatállomány, software disztribúció



Floptical® technológia a világ vezető hajlékony mágneses adathordozó gyártójától.

Az innováció Önnek dolgozik™

3M

3M Hungária Kft.
1133 Budapest, Váci út 110.
Tel.: (36-1) 267-1680, (36-1) 267-1683 Fax: (36-1) 267-1803

INFORMÁCIÓKÉRÉS: 26 ◄

Mindenki élete múlhat egy híváson!



A gyors intézkedést teszik lehetővé az M5-ös autópálya mentén elhelyezett segélykérő oszlopok. A hívó és a központ között rádiófrekvenciás kapcsolat van. A Közlekedési Minisztérium és az Autópálya Igazgatóság beruházási munkáit az ELIN végezte el.

ELIN

ELIN ELEKTRONIKA BUDAPEST KFT
1072 Budapest
Dob u. 54.
Tel.: 142-3734
Fax: 122-6423

INFORMÁCIÓKÉRÉS: 21

ALAPLAP

Mikroszámítógép magazin
mágneslemez melléklettel

Megjelenik havonta

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Munkatárs:
Sziebig Andrea

A Lemezkalauz
és a Közkincs szerkesztője:
Verebély Pálné

A szerkesztőbizottság tagjai:
Barna László, Boros György, Broczkó Péter, Brüll Károly, Farkas Ernő, Feleki Zoltán, Herczeg József, Kassay Árpád, Kónya László, Kovács P. Attila, Pintér Gábor, Vargha Dénes, Vékony Tamás, Villányi László, Zoltai Péter

Szerkesztőség, kiadó
és hirdetésszervezés:

1441 Budapest XI., Karolina út 17.
Telefon: 185-2421 Fax: 185-2221

Felelős kiadó:
Sebestyén Ilona
ügyvezető igazgató

Cédrus Kiadó Kft

Nyomdai előkészítés:
Tipoprint Kft, Budapest

Nyomtatás:
Zalai Nyomda, Zalaegerszeg
Felelős vezető: Galla József

Terjeszti a Magyar Posta. Előfizethető a hírlapkézbesítő postahivataloknál és a Posta Hírlapelőfizetési és Lapellátási Irodájánál (XIII., Lehel u. 10/a, Budapest 1900), vagy átutalással a 215-96162 pénzforgalmi számra.

Példányonkénti ár: 196 Ft
Évi előfizetési díj: 2 352 Ft

Külföldre terjeszti a Kultúra, Pf. 149, Budapest 1389

HU ISSN 0865-9788

Lapzárta után!

Sőt, már mostani számunk egy részének kinyomtatása után jött létre egy megállapodás, hogy kiadónk és szerkesztőségünk 10 hónap múltán, október 12-én ismét visszaköltözik a Cédrus Rt. Bp. XI., Karolina út 17. sz. alatti székházába. Postacímünk maradt az eddigi 1441 Budapest, Pf. 74, megváltozott azonban a telefonszámunk 185-2421-re, faxunk pedig 185-2221-re.

Nincs adat?

Ott és akkor nem is volt adat a számítógépekben, nem is fájt hát különösebben senkinek a szép, nagybetűs felirat. A két évvel ezelőtti választások bohózatba illően visszatérő refrénje azonban a számítógépet használó felől közelítve az egyik legfenyegetőbb rémkép. Ki ne hallott volna már olyat, hogy: „Egész nap csépelem a szöveget ebbe a ... gépbe, aztán meg elszáll minden"? Vagy a párja: „Végre rájöttem, hogyan lehet a legfrappánsabban megírni azt a rutint, de mire beírtam volna, az a ... portás lekapcsolta az áramot".

Az idézeteket a végtelenségig lehetne folytatni, hiszen az adatvesztés réme mindannyiunkat elér, pontosan mindig egyszer több alkalommal, mint amennyire pedagógiai megfontolásból szükség lenne.

Jó szokás szerint szerzőink helyenként most is vitatkoznak egymással, s nemcsak hangsúlyokban, hanem lényegi kérdésekben. Az az olvasó, aki tehát valami egyedül üdvözítő receptet vár az összeállítástól, bizonyos fokú csalódással jut majd a végére, mi viszont úgy véljük, olvasótáborunk többségét a mindig gyanakvók, a mindig és mindenből okulni vágyók teszik ki, s számukra minden információmorzsa, amely az adatok védelmében fogalmazódik meg, hasznosíthatóvá válik.

E havi kiemelt témánk tehát a „zabrala masina" effektusra koncentrál, úgy véljük, ha valaki csak egyetlen jó tippet talál is az összeállításban mindennapi munkájához, s ennek révén kinek-kinek fontos adatok őrződnek meg, az adott ember számára már mindenképpen megérte...

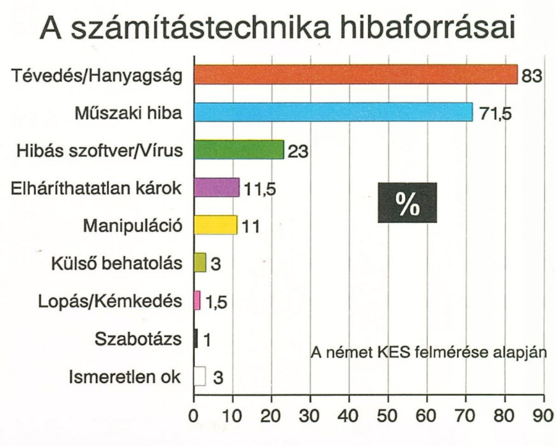
Adatvédelem, eszközvédelem

A számítógép nem kétfilléres eszköz.
Még akkor sem, ha eddig soha nem hallott alacsony árakon kínálnak gépeket.
Érthető hát, ha egyre-másra olyan ajánlatokkal találkozunk, amelyek a gépek fizikai védelmével foglalkoznak.

A kivehető, cserélhető merevlemez egyéb előnyei mellett biztonságot (adat- és eszközbiztonságot egyaránt) is jelenthet, hiszen páncélszekrénybe elzárható. A PC atyja, az IBM gépei mellé adott kézikönyveiben a biztonságnak sokkal nagyobb teret szentel, mint a legalább olyan fontos műszaki paraméterek ismertetésének. Még egy első pillantásra nevetségesnek látszó megoldás rajza is szerepel a PS/1 Pro kézikönyvében: az asztalhoz láncolható a gép.

Hasonlóan a gép fizikai védelmét lenne hivatott védeni a házzár, amennyiben azt valóban használnák is. Sajnos kevés példát látni erre. Az utóbbi hónapokban több kiegészítő hardveres védelemről is hallhattunk, olvashattunk. Kezdve a lemezegységbe dugható, kulccsal nyitható zárral, ami a lemezről indítást akadályozza meg, s folytatva a vírusellenes (TopGuard) és adat-/hozzáférés-védelmi kártyákkal (Farmosi István termékét kell megemlíteni). Ezeken kívül a Daxon kulcsos rendszere jelenthet megoldást — főleg nagy értékű hálózatok esetén, ahol létfontosságú, hogy illetéktelenül ne lehessen még bekapcsolni sem a gépet. Ez utóbbi megoldás egyedi kulcsokat alkalmaz, és így szükség esetén még az is naplózható, mikor, ki indította a gépet.

A 286-os gépeknél még nem nagyon gyakori, de 386SX és a feletti gépeknél már jelszó is beállítható a BIOS setupjában. Bár ez egyes típusoknál minimá-



lis szereléssel kijátszható, szükség esetén érdemes élni vele.

A szoftveres adatvédelem lehetséges megoldásai között is szerepel természetesen a jelszavas titkosítás, illetve hozzáférés-védelem.

A legjobb védelem persze az, ha a gépet arra használjuk, amire való: munkára. Igen nagy veszélyt jelent a gépre, és a benne tárolt adatokra, programokra, ha akár csak egyetlen játékot is beengedünk. Az első után jön második, a harmadik... s a végén előbb-utóbb jól fejlett vírussal leszünk gazdagabbak, s csak kínkeservesen tudunk megszabadulni tőle. Nem egy példát lehet erre idézni (lásd a Víruslélektan, Új víruslélektan és Vírushatározó című Alaplap Könyveket).

Hasonló veszélyt jelenthetnek állományainkra segítőkész, de hozzá nem értő kollégáink, akik jóindulatúan mellényúlva komoly károkat is okozhatnak. A géphez csak az nyúljon, aki tudja, mit csinál.

Még egy apróság az adatvédelemhez: az Egyesült Államok állami hivatalaiban csak másolásvédelem nélküli programokkal szabad dolgozni, hiszen a „védett" programoknál az eredeti lemez sérülése esetén rövidebb-hosszabb időre kiesik a gép a munkából, míg a nem védett programokat a másolatokról bármikor újra lehet telepíteni. Ezek után nemigen értem, mit akarnak a hazai másolásvédelmek gyártói. Alig hiszem, hogy ebben ők járnának amerikai kollégáik előtt. A védelemnek egyetlen helye lehet jogos: a bemutató példányok védelme a kiállításokon, a büntető jellegű védelem viszont az egyértelműen durva és büntetendő önbíráskodás sorába tartozik.

A biztonságot szolgálja az is, hogy a programokat megfelelő színvonalú, magyar (!) nyelvű kézikönyvekkel lássa el a forgalmazó. Sok „adaptált" program esetén a kézikönyvek használhatatlanok. Ezért nem árt, ha a programon kívül a mellé adott kézikönyvet is átnézi az a személy, akinek a megfelelő program kiválasztása a feladata. Egy jól dokumentált, kifogástalan online helppel ellátott programot akár egy „utcáról beszédült" ember is használhat néhány órás ismerkedés után. Amikor két programajánlat közül választani kell, döntő lehet a kézikönyvek használhatósága és a programok barátságossága közötti különbség.

A legnehezebb ügy a laptop és notebook gépek védelme, hiszen ezek igencsak könnyen mozgathatók. Ha a gépecskének kivehető merevlemeze van, legalább azt érdemes páncélszekrényben tartani, hogy a fontos adatok megmaradjanak egy esetleges betörés esetén.

Bár már sok szó esett róla, e helyütt is emlékeztetni kell arra, milyen nagy veszélynek teszi ki állományait az, aki alapos vírusellenőrzés nélkül vesz használatba bizonytalan „előéletű" lemezeket. Legalább egy, de ha lehet, inkább két keresőprogrammal is ellenőrizzük azokat a lemezeket, amelyeket nem mi használtunk utoljára saját, garantáltan vírusmentes gépünkben. Ehhez persze a víruskereső programok legfrissebb, legmegbízhatóbb verzióit kell használni. Hogy melyiket, az már ízlés és bizalom dolga. De ha egy külföldi eredetű programot választunk, célszerű mellette egy hazai keresőt is alkalmazni, hogy elkaphassuk a „hazai termés" képviselőit is.

Nagy Gábor

A megelőzés a legfontosabb

Dolgozni csak pontosan, szépen...

A napi munkához szervesen és elválaszthatatlanul hozzátartozik munkaeszközeink rendbetétele, karbantartása.
A számítógép mellett ez annyit jelent, hogy egyrészt programjaink, másrészt adataink biztonságáról is gondoskodnunk kell.

Az első szabály, amit nem szabad figyelmen kívül hagynunk, az, hogy — velünk ellentétben — „Murphy apánk" sohasem pihen. Ha akár a legkisebb lehetőségét is meghagyjuk annak, hogy a gép vegye át felettünk és adataink felett az uralmat, nem kétséges, előbb-utóbb meg is teszi. Így mindenkinek a figyelmébe ajánljuk az alábbi megelőző intézkedéseket.

— Csak ellenőrzötten vírusmentes gépen dolgozzunk. Ennek biztosításáról már előző számainkban is sokat olvashattak, így most nem kívánunk felesleges (?) ismétlésekbe bocsátkozni.

— Adathordozóink (merevlemez, floppylemez, streamerkazetta stb.) fizikai állapotát legalább havi, negyedévi rendszerességgel ellenőrizzük. Akár egyetlen meghibásodott szektor is nagy galibát okozhat. A hibás floppykat különítsük el a többitől, s — amikor időnk adódik rá — a megfelelő helyreállító, karbantartó programokkal (Chkdsk, NDD, PCTools stb.) próbáljuk meg rendbehozni őket. Az ismétlődően meghibásodó lemezterületeket jelöljük a DOS számára „bad sector"-nak, de jobb, ha egyszerűen kiselejtezzük a lemezt.

— Programjainkról legalább egy, de ha mód van rá, akkor két biztonsági másolatot készítsünk, és mindig csak a másolatról telepítsünk. Ha a program telepítésekor megsérül a lemez — ez néha előfordulhat —, akkor legalább nem a nehezen pótolható eredeti sérül meg.

— Másolásvédett programok használatát — ha egy mód van rá — kerüljük. Egyik-másik „büntető védelem" akkor is életveszélyes lehet, ha a program használata közben a jogosult felhasználó véletlenül mellényúl.

— Adatainkat rendszeresen, minden frissítéskor vagy legalább hetente, havonta archiváljuk. Ehhez ki-ki kedvenc és megszokott backup vagy tömörítőprogramjait használhatja. Fontos, hogy az adott programot megfelelően használjuk. Az archiváláson ezúttal kivételesen nem egy, hanem legalább két vagy három biztonsági másolat készítése értendő. A folyamat végén pedig ne felejtsünk el összehasonlító ellenőrzést (verify, test) kérni.

— Ne a DOS saját backup programját használjuk archiválásra. Sok bonyodalmat okozott, hogy verziófüggő, azaz az egyes DOS-változatok nem ismerik mindig a „testvér" formátumát, s így meglehetősen körülményesen sikerül csak visszaállítani az állományokat egy-egy véletlen gépmeghibásodás után. A másik ok, amiért érdemes más programokat használni: a DOS backupja lassú, nehézkes, és feleslegesen sok lemezt fogyaszt.

Adataink biztonságba helyezésére több megoldás is kínálkozik. Közöttük vannak olcsó és drága, egyszerű és bonyolult, lassú és szupergyors megoldások egyaránt. Az áttekintést táblázatunk tartalmazza.

Az ideális archiválási, adatmentési módszer megválasztása nem egyszerű dolog. Függ egyszer az adott feladattól, de ugyancsak függ a beszerzési és az üzemi költségektől, valamint a kezelhetőségtől. Ugyancsak sokat számíthat — de persze nem mindig — a sebesség.

Ha van rá mód, az egyik legjobb megoldás a cserélhető merevlemez. Gyors, a telepítést követő — ami néhány készüléktípusnál komolyabb szereléssel is járhat — használata egyszerű, nem igényel komolyabb műszaki ismereteket. Emellett, ha egy iroda mindegyik gépében azonos típusú cserélhető merevlemezt alkalmaznak, egy gép meghibásodása esetén gond nélkül

Hordozó	Megoldás	Előnyök	Hátrányok
Mágneslemez	Egyszerű másolás	Egyszerű használat, nem kell segédprogram, biztonságos	Lassú, nagy helyigény
	Backup programok	Egyszerű használat, gyors, automatizálható	Terjedelmes programok, kompatibilitási gondok
	Tömörítőprogramok	Lemeztakarékos megoldás	Keretprogram nélkül eléggé nehézkes
Merevlemez/ cserélhető merevlemez	Belső/külső	Az egyik leggyorsabb módszer, állományaink a lemezen azonnal hozzáférhetőek	Korlátozott kapacitás, egyszerre csak egy fér a gépbe, kissé drága az indulás
Írható-olvasható optikai lemez		Hosszú élettartam, nagy kapacitás	Drága meghajtóegység, lassú
Mágnesszalag	Streamer, beépített, vagy külső, hordozható	Nagy kapacitás, printerportról egyszerűen vezérelhetők, olcsó adathordozó, szabványos(ítható)	Szekvenciális tárolás

folytatható a munka egy másikon. A cserélhető vagy tartalék merevlemezre átmentett állományokat többnyire nem kell tömöríteni, azok azonnal hozzáférhető állapotban vannak.

Cserélhető merevlemezeket találhatunk például a Tandon (kétféle megoldással is), a Victor gépeknél, kaphatók SyQuest cserélhető merevlemezek és a Quantum ProDrive is megoldás lehet, bár kissé borsos a tárolóegységek ára.

A floppylemezekre mentés az egyik legáltalánosabban elterjedt archiválási módszer. Előnye a hordozhatóság — ha megfelelő programot használunk — és az egyszerűség. Hátránya, hogy lassú — néha kivárhatatlanul —, és hogy könnyű összekeverni a lemezeket, ráadásul a lemezek (főleg a régebbi típusú 5 1/4 inchesek) eléggé sérülékenyek. A nagyméretű állományokat lemezre másoláshoz vagy szeletelni, vagy tömöríteni kell, vagy mindkét módszert egyszerre kell alkalmazni. Megfelelő backup és/vagy tömörítőprogramokat alkalmazva olcsósága miatt — hiszen csak lemezt kell venni, a szükséges vezérlő és lemezmeghajtó már eleve a gépben van, s a szükséges programok is megfizethetőek — jelenleg a legáltalánosabb módszer.

Magyarországon jelenleg sem a magnetooptikai (MO), sem az írható-olvasható lézerlemez nem tartozik a komolyabban számításba vehető megoldások közé. Ebben főleg az ár és a lassúság a ludas.

A végére maradtak, de nem utolsó megoldásként a streamerek. A mágneses adatrögzítés egykor elsőszámú képviselői — bár kissé visszaszorultak a PC-kategóriában — a műszaki fejlődésnek és az árcsökkenésnek köszönhetően újra kezdenek felzárkózni.

Ma már széles választékban kaphatók különféle mágnesszalagos egységek. Vannak közöttük olyanok, amelyeket floppyként vezérel a gép, vannak, amelyek merevlemez-csatlakozót igényelnek, s vannak a laptopok bővítő rendszerbuszára csatlakoztatható készülékek is.

A legújabb — és talán legéletképesebb — verziók pedig a számítógépek párhuzamos (printer) portjáról vezérelhetők.

Ez a megoldás forradalmasíthatja az eddigi archiválási technikákat, hiszen egyetlen ilyen hordozható streamerrel és elegendő kazettával/szalaggal tetszőleges számú gép állományait helyezhetjük biztonságba anélkül, hogy meg kellene bontani a gépek házát. A legújabb típusok esetén még a rendszerindítást vezérlő CONFIG.SYS, és AUTOEXEC.BAT állományokat sem kell módosítani, elegendő a készülékkel adott szoftver használata. Nagy mennyiségű adat mentésére egyértelműen a mágnesszalagos/kazettás megoldás az optimális.

Nagy Gábor

Szalagarchívum

Ki ment ma még streamerre?

Sokáig keresgéltünk szerzőink között olyan szakembert, aki szalagos/streameres mentést használ. Szűk körű közvélemény-kutatásunk eredményeként rádöbbentünk — bár sejtettük —, hogy a DOS-világban honos programozóknak jóformán eszükbe sem jut az adatmentésnek ezt a formáját választani. A Unix-hívők táborában azonban helyenként — elsősorban anyagi megfontolásokból és csak „végszükség" esetén — használják a streameres mentést.

Ahhoz, hogy képet alkothassunk a szalagos mentési módokról, célszerű röviden áttekinteni a szalagok/streamerek alapvető tulajdonságait.

Ma már történelem

A mágneses adathordozók közül a szekvenciális elérésűeken, azaz a szalagokon az információ blokkok sorozatából áll, amelyeket csak egymás után tudunk elérni, sőt keresni is csak sorban tudjuk azokat. Ha valahova írunk a szalagra, akkor onnantól kezdve az összes mögötte levő tartalom eltűnik. Még ha azonos blokkhosszt is írunk, akkor is elképzelhető — sőt gyakran elő is fordul —, hogy a „mögöttes" információ elvész. Ugyanis a fizikai blokkok közötti szünetek (gapek) helyén indítja/állítja le a szalagmozgató motor a szalagot. Egyáltalán nem biztos, hogy a motor ott stoppol, ahol előző alkalommal. Az ilyen mechanikus gondok miatt könnyen előfordulhat, hogy a szalag további tartalmát ez a leállás „agyoncsapja".

A szalagon a hasznos információk olvasása a fizikai blokkok elején levő szinkronizációs bitek alapján történik. Ezen a többsávos információhordozón egy sáv egy bitpozíciónak (helyi értéknek) felel meg. A hasznos információkat különféle kereszt- és hosszparitásokkal védik (CRC, LRC). A védelem mellett ezekkel lehet még ellenőrizni a visszaolvasott információk helyességét is. A komolyabb szalagegységeknél azonban már íráskor (egy másik fejjel) visszaolvassák, hogy sikeres volt-e az írás.

A szalagegységek régebbi típusai közül az orsós szalagok (cserélhető szalagokkal dolgoztak) még ma is használatosak nagyobb gépeknél (800—6250 bit/inch írássűrűséggel).

A komolyabb „testvér"

A szalagegységeket mára felváltották a kazettás magnók, a streamerek. A különböző nagyságú és kapacitású kazetták közül a komolyabbak szabványos fizikai írásmódban (QIC = Quarter Inch Cartridge) dolgoznak. Ez a szabvány nemcsak a szalag méretére, hanem a fizikai írásmódra is vonatkozik: a különböző gyártóktól származó szalagegységek egymás „írásművét" is el

tudják olvasni. Ez azonban csak a fizikai olvasásra vonatkozik, de logikailag egyáltalán nem biztos, hogy meg is értik egymást.

A különböző kapacitású, írássűrűségű streamerek között ma már vannak olyanok, amelyek kapacitása akár a fél gigabájtot is elérheti. Az újabb, 8 mm-es DAT (Digital Audio Tape) szalagok gyakorlatilag a képmagnó elve szerint dolgoznak. Ezekre a kicsi, 8 mm széles szalagokra akár 2 GB is lementhető.

Gyakorlatilag (a fajlagos költséget tekintve) a QIC-szalag a legolcsóbb megoldás, akár a drive, akár a későbbi kazetták árát nézzük. A streamergyártók számára azonban rossz előjel, hogy rohamosan csökken a WORM-diszkek (Write Once Read Many) ára. Ezek nem szekvenciális elérésűek, így a mentett anyagok hozzáférése sokkal gyorsabb.

Mentőövek...

Érdemes kitérni a mentés szoftveres megoldásaira! Általában valamennyi hardvergyártó ad valamilyen szoftvert, amellyel tudunk szalagra menteni. Ismertek különféle mentőprogramok DOS alá, a Novell rendszerben pedig általánosan elfogadott megoldás, hogy egy munkaállomáson mentik DOS alatt az anyagokat.

A Unix-világban ismert klasszikus mentési eszközök (tar, cpio) mellett nagyon kényelmes és biztonságos megoldás, hogy a rendszerkomponensek és az installációs anyagok kazettán terjednek. (Néhány helyen ezek az anyagok már CD-lemezen is hozzáférhetők: Sun, Data General...)

Elmondhatjuk azt is, hogy általában minden System V. Release 4. tartalmaz olyan fejlett backup utilityt, amellyel intelligens módon megoldhatók a rendszeres mentések. Ehhez egy olyan tervet készítünk, amelyben megmondjuk, hogy mely fájlokat, milyen gyakorisággal akarunk menteni. A utility ezt automatikusan végrehajtja, csak a szalagot kéri hozzá. Ezeket a tevékenységeket természetesen adminisztrálja is, s mivel tudja, hogy a fájlnak mikori verziói vannak a mentőszalagon, csak akkor visz ki egy fájlt a mentendők közül, ha az azóta módosult. Visszatöltéskor pedig a szalag katalógusában logikailag másolhatunk, kijelölhetjük a visszahozandó fájlokat. Éppen ez a nagyszerű benne, hogy nem kell mindent másolni!

Persze ezeknél az eszközöknél sokkal megbízhatóbbak és gyorsabbak, ha olyan diszkalrendszert alkalmazunk, amely tökéletesen megbízható.

Gyakorlatilag

A szalagokról/streamerekről elmondottak ellenére — vagy ezeknek köszönhetően — nem terjed igazán a streameres mentés. Általában DOS alatt mindenki lusta streamerre menteni, nem beszélve arról, hogy egy ilyenfajta beruházás közel annyiba kerül, mint maga a gép (kb. 80 000 Ft). Minigépes környezetben is inkább WORM-ra mentenek, illetve egy megbízható diszkalrendszert használnak. Elterjedt azonban az a módszer is, hogy egy másik diszkre mentenek. Ez a megoldás már jól bevált az adatbázis-kezelőknél, ahol a naplózás egy másik diszken történik.

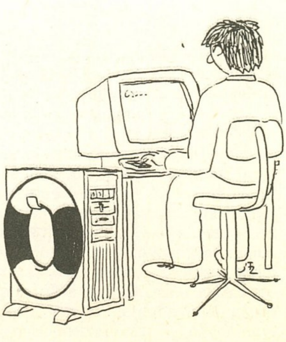
Sziebig Andrea

Keresd, míg megvan!

Rendszer(telen)táblázatok

A háttértári adatok elvesztése mögött a legtöbb esetben a rendszerterületen tárolt táblázatok meghibásodása áll. Ez azt jelenti, hogy az eltűnt adatok fizikailag továbbra is ott vannak valahol, valamilyen formában a mágneslemezen, csak az operációs rendszer a belső nyilvántartásainak sérülése miatt nem tudja megtalálni azokat...

A számítógépvírusok adatromboló tevékenysége is gyakran kimerül a rendszertáblázatok tökéletes összekeverésével. Az ilyen kórokozók nem radírozzák le az állományt a lemezről, mivel ez utóbbi módszerrel nem lehet igazán hatékonyan rombolni. Emiatt megelégszenek a partíciós táblázat, a FAT vagy a directoryadatok összekócolásával. Az ilyen tevékenységük olyan hibaüzeneteket vált ki az operációs rendszer részéről, amelyek hatására a gyakorlatlan felhasználó végleg lemond az odalett állományokról.



Amikor saját kezűleg kitörlünk egy fájlt, akkor sem történik más, mint hogy a DOS felszabadítja az állomány által elfoglalt mágneslemez-területet más adatok számára. A törölt adatok igazából csak akkor vesznek el, amikor azoknak a helyére valami mást írunk. Amíg ez meg nem esik, addig a megfelelő táblázat megfelelő helyén elvégzett nüansznyi módosítással a törölt adatok visszaállíthatók.

A fentiekből következik, hogy ha valamilyen módon időnként lementenénk az összes DOS-rendszertáblázatot egy mágneslemezre, akkor az esetleges serülés után azt helyre is tudnánk állítani. Talán az avatatlan felhasználó számára furcsának tűnhet, de a kérdéses

táblázatok együttes mérete elérheti, sőt egyes esetekben jócskán meg is haladhatja a 100 kbájtot. Azonban majdnem egészen bizonyos, hogy a lementésükhöz egy mágneslemez (legyen az akár csak 360-as) elegendő. Most nézzük meg részletesebben, milyen táblázatokról is van szó.

Kedvenc vírustanyák

Az IBM PC minden egyes winchesterén legfeljebb négy partíció kialakítására van lehetőség. A DOS szempontjából minden partíció külön háttértárolónak számít, a felhasználó úgy látja, mintha azok mind más-más winchesterek lennének. A partíciós táblázat adataiból a gép bekapcsolásakor működő program (a ROM BIOS start rutinja) megállapítja, hogy melyiken van a DOS operációs rendszer, és automatikusan erre a partícióra adja a vezérlést. Ha a partíciós tábla adatai valamiért megsérülnek, akkor a bekapcsolási rutin az „invalid partition table" szöveget írja a képernyőre, és nem tölti be a rendszert. (Előfordulhat, hogy a táblázatot egy vírus tette tönkre. Ebben az esetben esetleg más megtévesztő hibaüzenetet is kaphatunk. Ez külön említés nélkül vonatkozik a későbbiekben említett hibaüzenetekre is.)

Partíciós táblázat csak a winchestereken található, ugyanis a floppyk particionálását a DOS nem teszi lehetővé. Az összes többi táblázat mind a floppy-, mind a merevlemezeken rajta van.

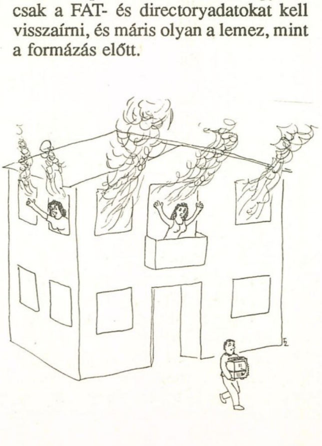
Ha a bekapcsolási rutin megtalálta a DOS-partíciót, akkor beolvassa onnan az úgynevezett bootrekordot, ami maga is egy kicsi program. Ennek a programnak az a feladata, hogy megtegye az előkészületeket a DOS rendszer felélesztéséhez. Sérülése esetén a „Disk boot failure, error loading operating system" üzenetet kapjuk. A bootrekord a vírusok kedvenc tanyája. A vírusok azonban olyan módosításokat hajtanak végre rajta, amelyek kezdetben észrevétlenek maradnak. A fertőződésről csak akkor szerzünk tudomást, amikor a lappangási idő letelte után a vírus rombolni kezd.

A következő DOS-táblázat a File Allocation Table, népszerűbb nevén a FAT. Itt tartja számon az operációs rendszer azt, hogy a mágneslemez mely területein melyik fájl található, és hogy hol vannak még szabad helyek. Amikor kitörlünk egy fájlt, akkor a DOS az addig elfoglalt területet szabadnak nyilvánítja ebben a táblázatban, de az adatokat nem törli le fizikailag a lemezről. Azok csak akkor vesznek el véglegesen, amikor valamit erre, a most már szabadként nyilvántartott helyre lementünk. Ilyenkor az új adatok felülírják a előző állomány maradványait. A FAT sérülése esetén a DOS a „file allocation table bad, drive <d:>" üzenetet írja a képernyőre.

A lemezen tárolt fájlok és tartalomjegyzékek neveit és egyéb adatait a DOS az úgynevezett directorytáblázatokban tárolja. Amikor elindítunk egy programot, vagy ha behívunk egy szöveges állományt a szövegszerkesztőnkbe, vagy egyéb más módon kívánunk használni egy fájlt, akkor a DOS ezekben a táblázatokban keresi a fájlnevet, és ha nem találja, akkor kapjuk a sokak által nagyon kedvelt „bad command or file name" üzenetet. Egy állomány törlésekor a FAT-táblázatban való helyfelszabadítás mellett a fájl nevének első betűjét is elállítja a DOS. Tehát a törlés után nemcsak a fájlban tárolt adatok maradnak érintetlenül a mágneslemezen, de a fájl neve is — igaz, ez utóbbi a kezdőbetűjét elveszíti.

Ha a törlés után visszaírjuk a fájlnév első betűjét a directorytáblázatba, és a fájlnak a mágneslemezen elfoglalt helyét jelölő FAT-bejegyzéseket is helyreállítjuk, akkor tökéletesen visszacsináltunk mindent.

Hasonlóan „felületes" munkát végez a format parancs is. Ez a FAT-táblázat minden egyes bejegyzését szabadra állítja, és a directory sorait kitörli. A sima fájltörléssel ellentétben nemcsak a fájlnév első betűjét törli, hanem a teljes directorybejegyzést! Itt is van mód az előző állapot visszaállítására, ugyanis csak a FAT- és directoryadatokat kell visszaírni, és máris olyan a lemez, mint a formázás előtt.



Bár ritkán fordul elő, hogy valaki „véletlenül" újraparticionál egy winchestert, de semmi sem lehetetlen, no és az ördög sem alhat állandóan. Ilyenkor a partíciós táblázat adatainak visszaállítása szükséges ahhoz, hogy újra a régi, megszokott bitjeikben pompázzanak adataink.

A fentebb említett táblázatokat csak szakavatott programozók módosíthatják sajátkezűleg, mivel felépítésük eléggé bonyolult ahhoz, hogy az egyszerű felhasználó ne matathasson eredményesen bennük. Szerencsére erre nincs is szükség, ezt a munkát elvégzik helyettünk az erre a feladatra készített segédpogramok.

A régebbi operációs rendszerek használói különböző utility programokkal tudták a DOS rendszerterületeit mágneslemezre menteni, hogy szükség esetén vissza lehessen azokat állítani. Az egyik leggyakrabban használt ilyen segédprogram a PCtools Mirror, de ezer más hasonló is kínálkozik a feladatra.

Az 5.0 verziószámot viselő DOS már saját berkein belül oldja meg ezt a feladatot. A táblázatok kimentésére szolgáló DOS programok az új rendszer gyöngyszemeinek tekinthetőek. Megjegyzendő, hogy e programok nem a Microsoft nevét dicsérik, hanem a PC Tools készítőjeként is ismert Central Point Inc. termékei. Nézzük most sorra ezeket a szerszámokat és szolgáltatásaikat!

Itt csak megemlítem a RECOVER parancsot. Ennek működése nem a rendszerterületek kimentésén alapul, és távolról sem vonatkozik rá a gyöngyszem jelző.

Mentegetősdi

Amennyiben a DOS adatbiztosítási szolgáltatásait rendszeresíteni kívánjuk, érdemes erre a célra egy megkímélt mágneslemezt kijelölni. Ezen a lemezen ne készítsünk alkönyvtárakat, tároljunk mindent a gyökérkönyvtárban. Első lépésben formázzuk meg a lemezt a FORMAT /S paranccsal, majd másoljuk fel rá a mirror.com, az unformat.com, az undelete.exe, a format.com és az fdisk.exe DOS programokat.

Igazi segítőtárs a Mirror nevű DOS program. Háromféle célra használhatjuk, kívánságainkat paraméterek közvetítésével tudathatjuk vele. A három szolgáltatás a következő:

1. A partíciós táblázat mágneslemezre mentése. Ehhez a /PARTN paramétert kell használnunk. Ekkor a program megkérdezi, hogy melyik meghajtóban lévő lemezre mentse a táblázatot. Válaszként elfogad winchestert is, de mindenképpen ajánlatos az erre a célra rendszeresített floppyra menteni. A művelet eredménye a kijelölt mágneslemez gyökérkönyvtárába helyezett partnsav.fil nevű állomány.

2. A bootrekord, a FAT és a root directory kimentése. Ehhez a művelethez a mirror d: /1 parancsot használjuk, a d helyére annak a lemezegységnek a betűjelét írva, amelyik táblázatait biztonságba kívánjuk helyezni. A megadott lemez táblázatait a gyökérkönyvtárba menti mirorsav.fil mirror.fil néven. A mirorsav.fil rejtett attribútumot kap, ezt a fájlt az egyszerű dir parancs nem listázza. Itt a mentés során nem adhatunk meg lemezegységet, a fájlok arra a lemezre kerülnek, amelyiknek a táblázatait tárolják. Ha igazán alapos munkát akarunk végezni, akkor a mirror.fil állományt másoljuk ki a floppylemezünkre. Ezt a másolatot nem tudjuk ugyan majd közvetlenül felhasználni, de végszükség esetén jól jöhet.

3. Fájltörléseket figyelemmel kísérő program ki/be kapcsolása. Bekapcsolásra a mirror /td parancs szolgál. A d betű helyére annak a lemezegységnek a betűjelét írjuk, amelyikről a törléseket naplóztatni kívánjuk. A kikapcsolásra az /u paraméter szolgál. A program bekapcsolása után minden egyes fájl törlésekor feljegyzi azokat az információkat, amelyek az esetleges visszaállításhoz szükségesek lesznek. A visszaállítás csak addig lehetséges, amíg az adott fájl volt területét a mágneslemezen egy másik fájl használatba nem veszi.

Restaurációk és restaurátorok

Amikor a katasztrófa bekövetkezik, akkor a mentéskor használt lemezről elindíthatjuk az unformat parancsot /partn paraméterrel. Ez megkérdezi, hogy melyik lemezegységben találja az előzőleg lementett partnsav.fil nevű állományt. A válaszunkat követően helyreállítja a sérült partíciós táblázatot.

A bootrekord, a FAT és a root directory visszaállításához még mindig ezt a floppyt használva kiadjuk az unformat d: parancsot. Ez akkor is megtalálja az előzőleg lementett táblázatinformációkat, ha időközben ki lettek törölve. Előfordulhat, hogy első nekifutásra nem találja meg azokat, de ha az ilyenkor felkínált

„Unable to find the MIRROR control file. If you want to search for the MIRROR image file through the entire hard drive, press Y, or press N to cancel the UNFORMAT command."
helyzetben igennel válaszoltunk, akkor át fogja kutatni a lemez teljes területét, és meg fogja találni a szükséges adatokat. A keresés megszakad akkor is, ha egy régebbi adatállományt talál. Az ilyenkor megjelenő
„A suspected MIRROR file starting at sector ##### has been found. The file is a backup to a more recent MIRROR image file. Do you want to use this file for unformatting or continue searching? Press Y to use this file, or N to keep searching."
üzenetre válaszoljunk az n billentyű lenyomásával, egészen addig, amíg a
„A suspected MIRROR file starting at sector ##### has been found. Do you want to use this file for unformatting or continue searching? Press Y to use this file, or N to keep searching."
üzenetet nem írja ki. Ekkor nyomjuk le az y billentyűt, ezután a program helyreállítja a kérdéses táblázatokat. Ekkor az utolsó mirror d: parancs kiadásakor aktuális helyzet állítódik vissza. Az azóta létrehozott fájlokat nem tudjuk helyreállítani. Ezért szükséges sűrűn menteni a rendszertáblázatokat. A mentés egy közepes teljesítményű gépen 3-5 másodpercet vesz igénybe, úgyhogy semmi akadálya annak, hogy akár félóránként elvégezzük.

Utoljára maradt a véletlenül törölt fájlok visszaállításának lehetősége. Erre a célra szolgál az undelete parancs. Kiadása után egyenként kiírja a törölt fájlok nevét a képernyőre, és mindegyikről külön megkérdezi, hogy helyre kívánjuk-e állítani. Ha a fájl helyét a lemezen egy másik állomány használatba vette a törlés óta, akkor a helyreállítás nem lehetséges. Erre az undelete parancs külön felhívja a figyelmünket.

Fridl György

A Make mint mentőeszköz

A programozók közül bizonyára sokan használják a Make segédprogramot, amelynek eredete a Unix-világba vezethető vissza. PC számítógépeken a Microsoft- és a Borland-verziók terjedtek el legszélesebb körben.

A Make program használatával sok forrásállományból álló programok kezelhetők egyszerűen, és ügyelve arra, hogy mindig a legújabb verziók kerüljenek lefordításra. Az egymástól való függőségek leírására egy makefile szolgál. A Make program a makefile-ban szereplő állományok létrehozási idői alapján határozza meg a tennivalókat.

Részletesebb leírással itt és most nem szolgálhatunk, inkább bemutatunk egy példát arra, hogyan is használható a Make arra, hogy szelektív mentést (azaz csak az utolsó mentés óta megváltozott állományok mentését) végezzünk. Ehhez a makefile-nak a következő formátumú bejegyzéseket kell tartalmaznia (minden egyes mentendő állományt külön le kell írni, és legalább egy üres sor szerepeljen a bejegyzések között!):

```
a:\mentes: c:\work\mentes
copy c:\work\mentes a:
```

A fenti leírás jelentése: az A meghajtón levő MENTES nevű állomány függ a C meghajtó WORK alkönyvtárában levő MENTES állománytól, e függőség pedig azzal oldható fel, hogy végrehajtjuk a következő sorban levő parancsot, azaz a C meghajtóról az A meghajtóra másoljuk az állományt. Ha azonban a Make program úgy találja, hogy a két meghajtón levő állományok keletkezési ideje azonos, vagy az A meghajtón levő az újabb, úgy a másolási parancs nem hajtódik végre.

Természetesen ugyanilyen egyszerűen megírható az a makefile is, amellyel a kimentett állományok visszaállíthatók.

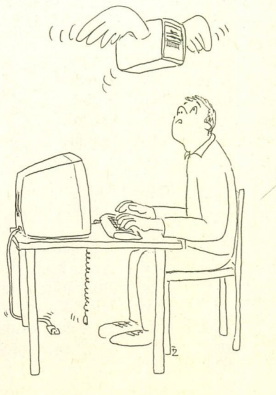
Hasonló célra szolgálnak a tömörítőprogramok update parancsai, valamint egyes backup programok is, azonban előnye ennek a megoldásnak, hogy a mentett állományok olvasásához semmiféle programra nincs szükség.

Ha el akarjuk kerülni, hogy bármilyen lemezre történjen másolás, akkor használhatjuk az Alaplap 1992. májusi számában megjelent Diskvol programot a lemez azonosítására. Ekkor a példa két sora közé szúrjuk be a következőt:

```
diskvol a: mentolemez
```

Ennek hatására csakis az A meghajtóban levő, mentolemez címkével ellátott lemezre történik a mentés, ellenkező esetben a Diskvol program által visszaadott kilépési kód félbeszakíttatja a Make programmal az illető állomány feldolgozását.

Lóth Tamás



Mentőszolgálat és elsősegély

Katasztrófák (backup) előtt és után

A legnagyobb körültekintés ellenére is bárkivel előfordulhat, hogy nagy becsben tartott adatait vagy programjait a technika ördöge lenyeli. Ha van biztonsági másolat (és nem ment az is tönkre), az a jobbik eset...

Az adatkatasztrófák — Murphy törvényének engedelmeskedve — sohasem közvetlenül a mentés után, hanem mindig pont az esedékes backup előtt egypár nappal vagy órával történnek. Ilyenkor aztán téphetjük a hajunkat, rohangálhatunk fűhöz-fához, hogy segítsen, továbbá megpróbálkozhatunk saját magunk is tenni valamit. Ha találunk olyan szakembert, aki némi térítés ellenében hajlandó rááldozni két-három napját a helyrehozatalra, odalett portékáink rekonstruálására (egyáltalán nem biztos, hogy sikerülni fog neki), a pénzt mindenképp elviszi. Van ugyan ezen a területen néhány (nem sok) profi, akikben megbízhatunk, de ritka fontos adatokat kell elvesztenünk ahhoz, hogy megérje foglalkoztatni őket.

A harmadik megoldás választásához szükségünk lesz egy nagy adag bátorságra és egypár jól használható segédszoftverre. Az ilyen célokra (is) használható szoftverek száma igen magas, mindenki megtalálhatja közte a pénztárcájának (uram bocsá' különleges beszerzési csatornáinak), vérmérsékletének és ízlésének megfelelőt. A bátorság nagyrészt az ismeretek függvénye, ezeket pedig nem árt időről időre összefoglalni.

Mielőtt bármilyen ügyeskedésbe kezdenénk, ajánlatos az adathordozón megmaradt romokat biztonságba helyezni, nehogy véletlenül tovább súlyosbítsuk a helyzetet. Ha floppy az áldozat, akkor ez nem túl nagy feladat, a diskcopy paranccsal készíthetünk másolatot, és ezután a másolat helyreállításával próbálkozhatunk. Ez teljesen kizárja a hiba további elmélyítését. Ha annyira súlyos a helyzet, hogy már a diskcopy sem hajlandó rendes munkát végezni, akkor kénytelenek vagyunk a próbálkozásainkat élesben csinálni. Ugyanez a helyzet winchester meghibásodása esetén is.

A rémségek

Az adatvesztést okozó hibákat két fő típusba sorolhatjuk. Az egyik ezek közül a fizikai meghibásodás (lásd az ábrát). Azt nem is nevezhetjük adatvesztésnek, amikor a kontroller vagy a floppymeghajtó mondja fel a szolgálatot, ilyenkor a tönkrement alkatrész cseréje után minden újból sínen van (hacsak utolsó rúgásként magát az adatállományt is tönkre nem tették a megbolondult áramkörök). Más a helyzet a winchestermeghajtó kiakadása esetén. Hiába sértetlenek az adatok magának a winchester lemezének a felületén, ha majdhogynem légmentesen el vannak zárva egy meghibásodott meghajtóban. Ilyenkor csak a szakszerviz segíthet.

A fizikai meghibásodások közül utoljára hagytuk a mágneslemez felületének rongálódását. Ha tartozik a lemezhez ingyenes adat-visszaállítási szolgáltatás, akkor vegyük igénybe. Ha nem, akkor is van valami esély a sikerre. A legfontosabb kérdés, hogymilyen fajta állomány van (volt) a rongálódott területen.

Ha .com, .exe, .bin, .sys, .ovl, .lib, .obj vagy bármilyen más kiterjesztésű bináris fájl a kárvallott, akkor kár kísérletezni, semmi esélyünk.

Ha egyszerű szövegállomány sérült meg, annak a megmaradt részeit a DOS recover <fájlnév> (vagy ha ez sikertelen, akkor fájlnév nélkül) parancsa a gyökérkönyvtárba menti
<file####.rec>
néven. Meg kell említeni, hogy a recover nem tartozik a DOS büszkeségei közé, csak végszükség esetén használjuk, és akkor is nagy körültekintéssel. Képes például arra, hogy a teljesen hibátlan állományokat is „megmentse", a végső őrületbe kergetve ezzel a jóhiszemű felhasználót. Ha rendelkezésünkre áll Norton nagymester legújabb utility-gyűjteménye, akkor a Disk Doctor (ndd.exe) sokat segíthet. Ez a program leteszteli a kiválasztott mágneslemezt, és ha valamit nem talál rendben, akkor kísérletet tesz a helyreállításra. A működése során bőbeszédű dialógusablakokkal informálja a felhasználót, választási lehetőségeket kínál, és javaslatokat tesz. Ha nem értünk az egészből semmit, akkor minden esetben fogadjuk el a felajánlott opciót. Ennek a procedúrának az eredménye a szintén a gyökérkönyvtárba helyezett file####._dd file, amely a tönkrement állomány megmentett részét tartalmazza. Ezt kell azután a kedvenc szövegszerkesztőnkkel tovább bűvölni, egészen addig, amíg ismét a régi pompájában nem látjuk ragyogni.

Az így helyreállított fájl ugyanis hiányos lesz. A recover és a Disk Doctor nem csinál mást, mint hogy kivágja az állomány károsult részeit, és csak a biztosan hibátlan darabokat írja be abba a bizonyos kimeneti állományba. A meghibásodott részt és annak környe-



zetét egyszerűen kiollózza, ezt nekünk kell a későbbiekben pótolni. A kiollózott szöveg általában nem több 500 vagy 1000 karakternél.

Ha a sérült fájl Lotus, Symphony vagy dBase formátumú adatállomány, akkor szintén Peter Norton tud segíteni rajtunk. A Filefix nevű segédprogram az ilyen adatbázisok helyreállítását végzi el.

Természetesen a sérült adatokat ez is kivágja az állományból, így ezeket újból rögzíteni kell. A Filefix ugyanolyan szószátyár, mint a többi Norton-program. Itt is tanácsos, hogy ha nem értjük valamelyik kérdést, akkor nyugodtan fogadjuk el az alapértelmezésként felajánlott választ.

A pusztítások

A logikai meghibásodások témakörébe tartozik a vírusok által elkövetett legtöbb gaztett, a véletlen fájltörlés vagy formázás, és a hibás programok rendszer-összeomlasztó vagy adatfaló tevékenysége is. Gyakorlatilag ide sorolhatjuk a rendszerterület fizikai meghibásodásait is. A vírusok igen lelkiismeretes munkát végeznek, az ő pusztításuk után már nem sok esélyünk lehet.

A véletlen fájltörlésből vagy formázásból eredő problémákra a régebbi DOS-verziókban a PC Tools Mirror, a Norton Utilities és számos más segédprogram jelentett gyógyírt. Az 5.0 operációs rendszer azonban már saját maga tartalmazza azokat a funkciókat, amelyek ilyen esetben segíteni tudnak. Természetesen a DOS 5.0 alatt is használhatjuk a régi, megszokott utility programokat, de az újfajta FAT-kezelés miatt a nagyméretű winchesterek mentésénél problémák merülhetnek fel. Természetesen az újabb Norton-szerszámok és a PC Tools friss verziói már megbirkóznak ezzel a problémával is. Tudni kell azonban, hogy míg a PC Tools és a Norton is csak a bootszektor, a FAT tábla és a gyökér-tartalomjegyzék adatait menti, a DOS parancsokkal lehetőség van a partíciós táblázat biztonságba helyezésére, valamint a mindennemű fájltörlés folyamatos naplózására is.

Fridl György

Nyugodt álmokat!

Mentés munka közben

A cím már sejtetni engedi, hogy nem valami falrengető dologról lesz szó, hanem valami olyasmiről, amit — triviális volta miatt — okos szakkönyvekben szinte illetlenség volna leírni, s amit mindenki jobbára a saját kárán tapasztal meg.
Meggyőződésünk: senki számára sem tanulság nélkül való számba venni a munkavégzés közbeni „balesetek" megelőzésének lehetséges eszközeit.

Elsőként vegyük az egyik legegyszerűbb óvintézkedést, a verify belső DOS parancsot. Célszerű már az autoexec.bat-ben egy verify on sort elhelyezni (az alapértelmezés verify off), így a DOS minden lemezre írás után ellenőrzi, hogy a lemezre írt adatok helyesen vannak-e rögzítve. Ily módon biztosíthatjuk, hogy az éppen felírt adat hiba nélkül olvasható legyen a későbbiekben is. Az ellenőrzés időt vesz igénybe, ezért a rendszer lassabban dolgozik, amikor a programok írnak a lemezre. Néha fontosabb a sebesség a biztonságnál, ilyenkor kikapcsolhatjuk az ellenőrzést, feltéve, hogy nem valamilyen „őskövület" winchesterrel (például ST 251) rendelkezünk.

Paraméter nélkül a verify parancs az aktuális ellenőrzési módot írja ki (verify is on/off). Az sem árt, ha a munkánkat némi winchester- vagy floppykarbantartással kezdjük, persze attól függően, hogy melyik médiumon kívánunk dolgozni, hiszen az idő pénz. Természetesen itt csak logikai karbantartásra gondolhatunk. Erre a tapasztalatok szerint a legmegfelelőbb segédprogram a Norton SpeedDisk 6.0-ás verziója. De miért is jó ezt használni? Minél többet nyúzzuk a merev-, illetve hajlékonylemezeinket (írás, visszaírás, törlés), annál inkább kezd „káosz" uralkodni rajtuk.

Ilyenkor találkozhatunk például a „lost chains ..." (elveszett láncolások) hibaüzenettel, és persze egyre növekszik az olvasási idő. Ez azért következhet be, mert az állományok nem folytonos blokkok egymásutánjában, hanem a háttértárolónk teljes felületén tördelve, összevissza helyezkednek el. Így a DOS-nak egyre több láncolási címet kell tárolnia, s persze ezzel növekszik a hibázás valószínűsége is betöltéskor.

Olvasáskor a fej pedig összevissza pozicionál a lemez fölött — keresvén az olvasandó állomány darabkáit —, ezzel is növelve a hozzáférési időt.

Az előbb elmondottak természetesen érvényesek a hajlékonylemezekre is, de még fokozottabban a kisebb kapacitás, illetve a nagyobb hozzáférési idő miatt. A Norton SpeedDisk tehát kiválóan alkalmas a háttértárolónk optimalizálására. Ez a következőkből állhat: az állományok és az üres hely szétválasztása, a széttördelt állományok folytonos blokkok egymásutánjában történő elhelyezése, az állományok fizikai sorrendjének módosítása, illetve a könyvtárszerkezet újraszerkesztése a gyorsabb elérhetőség érdekében. A program elindítása után közli, hogy az adott háttértároló hány százalékán nincsenek széttördelve az állományok (pl. 95% of Drive C: is not fragmanted), és javasol egy optimalizálási eljárást (Recommended optimization method) az előbbi érték alapján.

Általában válasszuk ezt, de nagyobb munkáink előtt, ettől függetlenül, válasszuk a teljes optimalizálást (Full optimization) a főmenü Optimize / Optimization method almenüjéből. Megbízhatóan és gyorsan dolgozik. Egy 286/16 MHz-es AT-n legrosszabb esetben is 10 perc alatt végez. A program még sok érdekes funkcióval rendelkezik, melyek e cikk határain kívül esnek,

így csak javasolható, hogy mindenkinek megéri egy délutánt eltölteni e remekbe szabott program társaságában. Akik azonban installálták a Stacker 2.0-t, már csak a hozzá adott SDEFRAG-ot használhatják, amely hasonló tulajdonságokkal rendelkezik, mint az előző, csak sokkal lassabb a tömörítés miatt. A korábban említett géptípuson a legjobb esetben is 30-40 percet dolgozik. Az SDEFRAG-nál így érdemes mindig a javasolt eljárást választani. Tapasztalataink szerint a Stackerrel „tuningolt" háttértárolókon a széttördelt állományok aránya úgy is 10% alatt van.

Ez után a hosszúra nyúlt felvezetés után rátérhetünk a lényegre. Nagy általánosságban a következőket kövessük mentéseink során: 10-15 percenként — vagy legalább a nagyobb változtatások után — mentsük el műveinket, integrált fejlesztőrendszerek esetében legalább a futtatás előtt mentsünk, főleg ha egy „jól irányzott" assembler rutint szeretnénk kipróbálni. Ezenkívül még sok egyéb beállítással könnyíthetjük meg a dolgunkat. Úgy véljük, érdemes megnézni egy konkrét példát.

Vegyük a Turbo Pascal különböző változatait a 4.0-ás verziótól a Turbo Pascal for Windowsig, hiszen valamelyik verziójával a legtöbben találkoztak már. Az 5.5-ig az Option/Environment almenüpontban a következő beállításokat tehetjük: Backup source file on, .BAK fájl képzésére mentéskor, az előző állapot megőrzésére; Edit auto save on, automatikus mentés, ha a Turbo rendszerből kilépünk; Config auto save on, a konfigurációs fájl automatikus mentése.

A 6.0-ás verzióban már jelentősen bővültek az effajta lehetőségek is. Először az Option/Environment/Preferences almenüben az Auto save-nél kapcsoljuk be mindhárom opciót. Az Editor files bekapcsolásával a szerkesztett fájl vagy fájlok, az Environment (környezet) opció bekapcsolásának hatására az integrált fejlesztői környzet (IDE = Integrated Development Environment) jellemzői, a Desktop bekapcsolásával pedig a munkaállományok és azok környezetére vonatkozó jellemzők mentődnek el minden fordítás+futtatás és kilépés előtt, így az IDE elvégzi helyettünk a szükséges mentéseket. Ezután lépjünk be az Options/Environment/Editors almenübe, ahol most számunkra a következő opciók érdekesek: Create backup files, amiről már volt szó az előzőekben, illetve a Use tab characters. A Use tab characters bekapcsolása esetén a Turbo Pascal ASCII 9 tabulátort tesz a forrásfájlunkba, ha a TAB-ot nyomtuk meg, különben üres helyek kerülnek bele, így némi helyet takaríthatunk meg.

Ha már itt vagyunk, nyugodtan bekapcsolhatjuk a többi opciót is, ha igazán kényelmesen szeretnénk dolgozni. A Turbo Pascal-változatok közül még a Windowsos verzió maradt hátra, amelynél az összes eddig említett opciót az Options/Preferences almenüben állíthatjuk be.

Ezekhez hasonló lehetőségeket sok más programnyelv integrált fejlesztőrendszerében és a különböző szövegszerkesztőkben is találhatunk, így ott sem fog különösebb gondot okozni a beállításuk.

Végezetül szólnunk kell egy — ma még nem eléggé — közismert magyar közmondásról, mely szerint: ki mint készít másolatot, úgy alussza álmát. Bármennyire is minőségiek a háttértárolóink, mégis előfordulhat, hogy például egy „kölcsön kapott" vírussal tönkretehetjük adatállományainkat. Tehát fontosabb műveinkről mindig készítsünk legalább egy másolatot, a gyengébb idegzetűek készíthetnek kettőt is!

Erre legegyszerűbb megoldásnak a BACKUP/RESTORE külső DOS parancspár használata kínálkozik, de persze nyugodtan használhatunk valamilyen tömörítőt is. A BACKUP egy vagy több fájlt archivál egyik lemezről a másikra, ahol a meghajtóknak különbözőeknek kell lenniük.

Használata a következő:
BACKUP [forrásmeghajtó] [útvonal] [fájlnév] [célmeghajtó] [/S].

A /S választható opció hatására a katalógusban lévő alkatalógusban elhelyezett fájlokról is készül másolat.

E parancs párja a RESTORE, mely egy BACKUP-pal készített archivált fájlt állít vissza az egyik lemezről a másikra. Használata a következő:

RESTORE [forrásmeghajtó] [célmeghajtó] [útvonal] [fájlnév] [/S].

Ha az útvonal helyére *.*-ot írunk, az összes fájl visszaállítódik. Ha valaki még ezután sem volna képes nyugodtan aludni, használhatja a CHKDSK szintén külső DOS parancsot másolatának ellenőrzésére, amely végignézi a megadott lemezen lévő katalógusok, fájlok és a helyfoglalási tábla (FAT) helyességét, s erről jelentést készít fájlba vagy a képernyőre. A /F opcióval utasíthatjuk a CHKDSK-et, hogy az esetleg észlelt hibákat javítsa ki. Akinek ez sem volna elég, annak ajánlom a Norton Disk Doctor 6.0-ás változatát. Remélem, mindenki nyugodtan fog aludni.

Kardos Balázs

Tömörítők és adatbiztonság

Zanzátlanítás — apróbb zökkenőkkel

Mindmáig vannak olyanok, akik idegenkednek a tömörítők alkalmazásától, mondván: „Mit tudom én, mi lesz az adataimból, ha kicsomagolom őket?" Vajon fenyeget-e az adatvesztés réme ezen a területen? Szerzőnk egy kis „leltárt" is készített a jelenleg elérhető tömörítőkínálatról.

Ha tömörítőprogramokkal oldjuk meg adataink biztonságba helyezését, három dolgot mindenképpen szem előtt kell tartanunk.

1. Egyes vírusok előszeretettel vadásznak ARC és ma már ZIP állományokra is. Ha már megszoktuk választott tömörítőprogramunk használatát, és nem akarunk másikra átállni, legalább azt az elővigyázatossági intézkedést tegyük meg, hogy egzotikus kiterjesztésekkel védekezünk ezek ellen a kártevők ellen. Ez a megoldás valamennyire véd az illetéktelen hozzáférés ellen is, így nem kell titkos anyagainkat jelszó megadásával elrejteni — a jelszavakat közismerten hamar elfelejti az ember, és akkor fuccs az adatoknak.

2. Archív állományainkat minden módosítás után (ARJ-nél közben is lehet) ne felejtsük el ellenőrizni. Lassítja a munkát, de a biztonság mindenképp megéri. Meglehetősen kínos ugyanis, amikor egy elmentett állományt nem tudunk az archív sérülése miatt visszaállítani.

3. Tartsuk szem előtt, kinek is készítjük a tömörített állományokat. Bosszantó ugyanis, ha az archív címzettje nem tudja kibontani a neki szánt állományokat, mert nem ismeri a szükséges kibontó programot, vagy a szükséges kibontó nincs kéznél. Ilyen esetekben célszerű lehet önkicsomagoló (SFX) archívokat készíteni, vagy a kibontó programot is az archív mellé másolni az utazó lemezre a hozzá tartozó magyar nyelvű használati utasításokkal, vagy egy előkészített, automatikus kirámolást vezérlő batchállománnyal.

Adattömörítő eljárásokat ma már egyre több program használ. A valódi, elsődlegesen adattömörítési céllal készített programokon kívül a grafikus állományokat kezelő, a táblázatkezelő és a kommunikációs szoftverek jó része is rendelkezik valamiféle adattömörítési lehetőséggel. A mindennapi használatban megszokott DOS keretprogramok újabb verzióit is ellátták már a legelterjedtebb tömörítők állományait kezelő funkciókkal. A PC Tools, az XTree és a Norton Desktop for DOS, Norton Commander (lásd Zipview) már felkészült egyes tömörítési eljárások, programok használatára.

A csaknem klasszikusnak számító tömörítők újabb és újabb verziói hol ügyesebben, hol kissé ügyetlenkedve szolgálják az adatbiztonságot.

Megjelent a Hyper nevű tömörítő 2.60-as verziója. Azonkívül, hogy német nyelvű lett a program és a leírás, s már csak egy szerző (Sawatzki) jegyzi a programot, nemigen találni más változást. Továbbra is jól használható a számolótáblák zsugorítására, s a .HYP kiterjesztés is jelenthet némi védelmet részint a vírusok, részint az illetéktelen hozzáférés ellen.

A PKZIP új, 2.0 verziószámot viselő verziójának még csak a híre gyűrűzött be hozzánk, néhány, 2.01-nek csúfolt kalózpéldányt viszont már „sikerült" beszerezni, de ezektől Phil Katz mereven elhatárolta magát. Mikor a programokat és leírásaikat áttanulmányoztuk, egyértelműen kiderült: Katznak minden oka megvolt rá, hogy így tegyen. Ez nem az a megszokott színvonal, ami a PKWare programokat jellemzi.

Az utolsó rendelkezésünkre álló, érvényes Katz-verzió az 1.93alfa. Bár a szerző állítása szerint előfordulhat adatvesztés a program használata során — hiszen kellően ki nem érlelt technikát alkalmaz —, nekünk eddig nem sikerült semmi ilyet produkálnunk vele. Egy dolgot mindenesetre érdemes megjegyezni. Ha lecseréljük a PKUNZIP program 1.10-es váltózatát az 1.93alfára, nem lesz ugyan gondunk az új ZIP-pel készített archívokkal, de elveszítjük az autenticitásvizsgálat lehetőségét. Ez egyes archívoknál fontos lehet.

Yoshi mester változatlanul nem adott ki újabb verziót közkedvelt LHA programjából. A hírek szerint most azon dolgozik, hogy egyszerre több platformon (Unix, OS/2, DOS stb.) legyenek egymással adatszinten kompatibilis tömörítők.

Az ARJ nagy változáson ment keresztül. A bővülések egyike a ZOO-hoz hasonló többgenerációs adattárolás lehetősége. Ez lehetővé teszi, hogy a programfejlesztők megőrizzék az egyes fázisok állapotait. Ugyancsak a biztonságot érinti a /jt kapcsoló bővülése. Ha /jt1 formában használjuk, akkor a normál CRC-ellenőrzésen felül egy teljes állománytartalom-összehasonlítást is végez a program. A Teszt parancs mellett a báziskönyvtárat is fel tudja használni, és az archívban őrzött (teljes) neveket is az állományok ellenőrzésére. A bepakoló és kipakoló parancsoknál csak az archívban tárolt nevekkel dolgozik. A /jt2 ugyanazt végzi, de csak az éppen frissen bepakolt állományokat ellenőrzi, így kerülve el azt a kellemetlen helyzetet, amikor a program azért sipákol, mert egyes, az archívban lévő állományok nincsenek meg az aktuális vagy a báziskönyvtárban.

A Stacker 2.0 verzióban is van néhány érdekesség. A telepítéskor az ismerős Norton SpeedDisk programot láthatjuk a Stacker telepítőprogramjai között. A programcsomag része az Sdefrag program. Ennek használata a várakozással ellentétben nem növelte, sőt csökkentette a szabad területet. Ejnye! Ez a modul még fejlesztésre szorul. Más vonatkozásban pedig meg voltunk elégedve a programmal, hiszen megbízhatóan működött.

Egyébiránt a Stac Electronic jelenleg programjának és kártyájának OS/2 alatt használható, mikrocsatornás változatán dolgozik. Kíváncsian várjuk, mikorra lesz belőle termék. A másik, Stackert illető újdonság, hogy a béta-tesztelés alatt álló MS-DOS 6.0 tartalmazni fogja a Stacker egy verzióját. Ha az

ellenfél lépett, a Microsoft sem maradhat le.

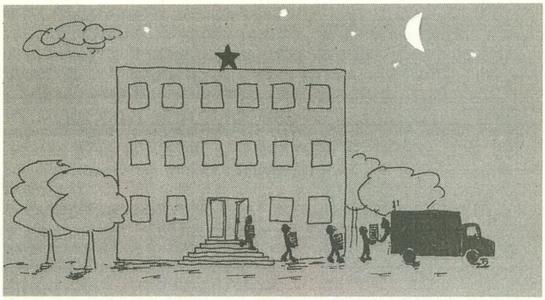
A Windows 3.1 felhasználói azonban mondjanak le a lemezduplázó programok használatáról, hiszen az új Windows-verzió „önfejű" lemezkezelése komoly lefagyásokat, adatvesztéseket okozhat.

A Stacker egyik komoly vetélytársa, a SuperStore sem mentes az apróbb hibáktól. Volt, akinek komoly adatvesztésébe került az a felismerés, hogy a program a felpumpált könyvtárakban csak korlátozott számú (256) bejegyzést engedélyez, s erre nem figyelmeztet.

Egy új lemezduplázóról is érdemes beszámolni. Pár hónapja jelent meg a nyugati szaklapok hirdetéseiben az IIT terméke, az Xtradrive. Az eddig inkább hardvereikről (aritmetikai koprocesszor) ismert cég fiatal fejlesztőgárdájának terméke már az 1.0 verzióban sem nagyon marad le a Stacker és a SuperStore mögött. Tömörítésben jobb azoknál, de az adatbiztonságon van még mit javítani. Az Xtradrive segítségével felpumpált lemezt teljesen átírja a program, így egy esetleges vírusfertőzés esetén komoly gondok jelentkezhetnek.

Hasonlóképp veszélyes, ha véletlenül letöröljük az indításhoz szükséges rendszerállományokat. Ilyenkor csak egy speciálisan erre a célra előkészített, az Xtradrive meghajtóját is tartalmazó rendszerlemez segíthet.

Nagy Gábor



Elektronikus aláírás és titokmegosztás

A jövő adatvédelmi módszerei

A két egymásra épülő rendszer számítógépben tárolt adatokat, programokat véd meg illetéktelen adatbevitel, adatmódosítás és adatvisszanyerés ellen, valamint nagyon hatékonyan alkalmazható vírusfertőzés megelőzésére.

Az üzenethitelesítő azonosító (MAC: Message Authentication Code) lehetővé teszi a tárolt vagy átvitelre kerülő adatok elektronikus aláírását. Ezzel az aláírással a beavatkozás időpillanatában — real time módon — azonosítható a tárolt állományért vagy az üzenetért felelős személy. Az elektronikus aláírás megvédi az állományt és az üzenetet az észrevétlen változtatástól, hamis adatok bevitelétől.

Növeli a hitelességet

Ez a hitelesítési rendszer különbözik a már ismert rendszerektől, például az ISO/DIS 8737-2 nemzetközi szabványtervezetben leírtaktól. Bár még csak nemzetközi szabványtervezet formájában van meg, de már láthatók alkalmazásának, illetve hatékonyságának korlátai. Az aláírás matematikai háttere a latin négyzetek és a permutáló polinomok elméletén alapul. A rendszernek több előnye van az ismert rendszerekkel szemben, lehetővé teszi például a többszörös aláírást, amit a mindennapi gyakorlatban a hitelesség növelésére használnak az üzleti levelezésben. A rendszer segítségével bármilyen szerkezetű, méretű (akár változó hosszú) és összetételű rekord, állomány, vagy akár üzenet aláírható.

A titokmegosztás

Titokmegosztáson valamilyen cselekmény, tranzakció lebonyolításához szükséges információ több személyhez kapcsolását értjük. Ez azt jelenti, hogy egy bizonyos tranzakció akkor és csak akkor következhet be, ha adott számú egyén van jelen egy adott helyszínen, egy időben. A rendszer a banki páncélszekrény kinyitásakor alkalmazott eljárás elektronikus változatának tekinthető. Ugyanis a páncélszekrény kinyitásánál a bank képviselőjének és az ügyfélnek egyidejűleg kell jelen lennie. Mivel a páncélszekrénynek két zára van (különböző kulcsokkal), az egyik kulcs az ügyfél, a másik a bank birtokában, a páncélszekrény pedig csak a két kulcs egyidejű használatával nyitható ki.

A titokmegosztási rendszer például az alábbi két módon használható.

A számítóközpontokban a számítógépes bűncselekmények legtöbbje akkor következik be, amikor a gépteremben csak egy operátor van jelen. A titokmegosztási rendszerrel olyan új ki- és belépési ellenőrző rendszer valósítható meg, amellyel biztosítható a számítógép működésének korlátozása, leállítása, ha a kezelők száma egy meghatározott érték alá csökken.

Jelenleg az aktív memóriakártyákat kulcsszóval (ún. PIN-szám) védik. A kártya felhasználójának azonosítására ez a módszer nem vált be, a rendszer

gyenge pontjának bizonyult. A számítógépes bűnözést elemzők döntő többsége kimutatta, hogy több lehetőség is van a kártyák kulcsszavának illegális megismerésére, illetve a kártya használatára. Ellophatják őket, és teljes kipróbálással megállapíthatják vagy lehallgathatják. A fentiek miatt a bankok — védve saját érdekeiket és igyekezve minimalizálni kockázatukat — korlátozzák az egy tranzakcióval felvehető pénz maximális összegét. A titokmegosztás lehetővé teszi, hogy a tranzakció végrehajthatósága két vagy több személytől függjön, akiknek a tranzakciónál egy időben kell jelen lenniük. Ebben az esetben egyetlen tranzakcióval lényegesen nagyobb összeg hívható le a bank kockázatának minimálisra csökkentésével, mert a tranzakciónál több kezes van jelen.

Miért ez a kettő?

A jelenleg használt védelmi rendszerek az egyre kifinomultabb és szakszerűbb számítógépes bűnözéssel szemben mind tehetetlenebbnek bizonyulnak. Ezért áll jelenleg kidolgozás alatt egy nemzetközi szabvány (ISO/DIS 8737-2), amelyről azonban már most látható, hogy csak lassítja a számítógépes bűnözés által okozott károk növekedését, de igazán hatékony védelmet nem nyújt.

Ezzel szemben a két rendszer az említett szabvány védelmi hatékonyságán messze túlmutat, és jelenleg elképzelhetetlen olyan módszer, amellyel feltörhető lenne. A két rendszert menedzselő cég 50 000,— Ft-ot ajánl fel az elektronikus aláíráson alapuló védelmi rendszer feltörőjének.

Hol lehetne a legjobban alkalmazni ezeket a rendszereket?

Magyarországon már napirendre került a korszerű európai jogalkotásnak megfelelő adatvédelmi és népességnyilvántartási törvény megalkotása. E törvények hatékony védelmi rendszerek kiépítése nélkül nem hajthatók végre. Másrészt a pénzügyi szféra egyre veszélyeztetettebb helyzetbe kerül — különösen a forint konvertibilissá tételekor — a jelenlegi bankbiztonsági rendszerek mellett. Tehát legfontosabbakként a pénzügyi szektor, az államigazgatás és a nemzetvédelemi szektor merülhet fel mint potenciális alkalmazó.

Vírus ellen is aláírást!

Mint a bevezetőben említettem, az elektronikus aláírás önmagában is, de a titokmegosztással együtt különösen alkalmazható vírusvédelem céljaira. A védekezés alapvető gondolata a megelőzés az elkésettnek mondható detektálással és irtással szemben. Az aláírás segítségével a hardvertől az operációs rendszeren át az adatállományokig minden aláírható, hitelesíthető. Az aláírással megakadályozható az állomány, rendszerterület módosítása, és megállapítható a behatolási kísérletet tevő személye. A titokmegosztás segítségével előírható az egyes adatok, programok elérhetőségének és használatának szigorú köre.

Nem lehet elégszer hangsúlyozni, hogy egy vírusos állomány milyen hatalmas károkat okozhat a magyar monetáris rendszerben, és beláthatatlan következményekkel járhat a nemzetvédelem számítóközpontjaiban.

Tass Csaba

szoftver ABC Kft.

☎ : 201-6891
201-2011/131
: 201-8619
✉ : 1277 Budapest
23. Pf: 45.

Rövid határidővel szállított szoftvereink:

(Ár ÁFA-nélkül!)

Ariadne 2.1 (Cldec87)	49.900	MS Word for Windows	46.000
DOSHun	6.000	MS Word for Windows Multispeller	7.600
Ékszer	19.900-45.000	MS Word for Xenix 386 / Unix 386	95.000
Hot Line	19.900	Nantucket Tools II (angol)	55.000
HunHyp	10.000	Nantucket Tools II magyar kézikönyv	2.000
Kontír 2000	22.000	Netroom Single User	9.900
Lektor	15.000	Netware Lite	8.500
Napló 2000	7.900	Netware SQL	51.000
WinHun	6.000	NewsMaster II	8.900
WordPerfect (magyar)	37.000	Norton Anti Virus	10.800
		Norton Backup	8.300
		Norton Backup for Windows	8.300
allCLEAR	26.000	Norton Commander	12.400
Adobe TypeManager	10.500	Norton Desktop for Windows	14.800
Aldus Pagemaker 4.0	**49.900**	Norton Editor	9.900
Ami Professional	46.000	Norton Utilities	14.500
Anti Virus +	14.900	Object Vision	15.000
Blue Max	12.900	On Target	33.100
Borland C++	30.000	On Track Disk Manager	9.000
Borland C++ & Appl. Fram. 3.0	42.900	OrCad PCB	198.000
Carbon Copy for Windows	20.000	OrCad VST	163.100
CC:Mail Fax	218.000	Paradox	37.500
CC:Mail Gateway	142.000	PC Anywhere IV	14.900
CC:Mail Remote	35.500	PC Astro	9.000
Charisma	47.000	PC Cosmos	7.900
Checkit V3.0 /Hardware-Diagnos./	13.900	PC Globe	8.500
Chiwriter Professionell	42.000	PC Paintbrush IV Plus	18.900
Clarion Profess. Developer	78.000	PC Tools 7.1	13.500
Clipper 5.01	**62.000**	PerForm Pro for Windows	41.500
Corel Draw 3.0	**26.000**	Personal Rexx	19.000
Crosstalk for Windows	18.000	PharLap 386 / VMM	27.500
dBASE IV 1.5	13.900	PhotoStyler	74.000
DBFast for Windows	42.000	PopDrop Plus	11.000
Designer	54.500	Presentation Team	44.900
Deskview 386	21.500	Printer Assist	27.000
Desqview Qemm 386	14.000	Printshop	7.500
Desqview QRam	9.900	Procomm Plus	13.000
dGE	25.000	Publishers Paintbrush Win. 3.0	40.500
DR DOS	8.500	Publishers Type Foundry	45.900
Draw Perfect	35.000	Q & A	37.000
Draw Plus	13.000	Q Assist	21.000
Easyflow	19.900	Quattro Pro	14.800
F & A	49.500	Quicksilver	46.500
Fontasy	12.000	R & R Rel. Report Writer	24.000
Forest & Trees	49.000	**Recognita**	**98.000**
FoxPro	61.490	SCO Unix 3.2 Dev. Pack	88.000
FoxPro LAN	**70.000**	SCO Unix 3.2 Oper. Sys.	88.000
FoxPro Toolbox	59.000	SCO Foxbase Plus 386	69.000
Framework IV	55.500	Show Partner FX	31.500
FreeHand	56.900	Show Partner Picture Pack	22.000
Go Script Plus	26.000	Sideways	14.500
Grammatik IV for Windows	12.500	Sit Back for Windows	15.000
Gupta Quest for Windows	69.500	Smalltalk V	12.900
Gupta SQL Base Sigle User Dos	61.000	Smalltalk V Windows	36.000
Gupta SQL Windows for btrieve Lan	17.000	Smarterm 320	18.500
Halo Windows Toolkit	52.500	Soft Type	16.500
Harvard Graphics	57.000	Software Bridge	13.900
Harvard Graphics for Windows	49.900	Software Carousel	12.000
Hijaak	19.900	SpeedStor	12.000
Image Prep for Windows	35.000	SPSS/PC+ Base	52.000
Just Write	16.600	SPSS/PC+ Statistic	45.000
K-Edit	17.500	SPSS/PC+ Advanced Statistic	45.000
LAN Assist Plus	36.000	SPSS/PC+ Graphic. Int.	30.000
Landmark Speed Test	6.900	Statgraphics	78.000
Laplink Professional	16.000	Superbase IV	62.000
Lexica	27.000	Superbase IV Lan	115.000
Lotus 1-2-3 for Windows	55.000	Time Line	58.000
Major BBS 2 line	19.000	Turbo C++ Windows	15.000
Managing your money	14.500	**Turbo Pascal Professional**	**12.500**
Map Assist	37.000	**Turbo Pascal for Windows**	**10.200**
MathCad for MS Windows	46.000	Ventura Publisher 4.0 Win	74.000
MathType for Windows	28.500	VM / 386 Multiuser	69.000
Matrix Layout	24.000	WinConnect	11.500
MS C ++ 7.0	**39.000**	Window Base	49.000
MS DOS 5.0	**4.200**	Windows CAD 2D for Windows	79.000
MS Excel	**43.000**	Windows Maker Prof.	73.000
MS Macro Assembler	11.900	Winfax Pro	15.000
MS Office for Windows	81.000	Wingz for Windows	54.900
MS Pascal	26.000	Wordperfect 5.1	37.000
MS Quick C for Windows	18.000	Wordperfect for Windows	37.000
MS Visual Basic	18.000	**Wordperfect Works**	**12.500**
MS Windows 3.1	13.800	Wordstar 6.0	36.000
MS Windows Entertainment Pack	5.500	XTree net Advanced	55.900
MS Word 5.5	37.000	Zinc Interface Lib. 2.0 Borland	39.000
MS Word 5.5 Multispeller	12.000	Zortech C++ Developers Ed. V3.0	58.000
MS Word Exchange	7.200		

Hát persze,
hogy a többi szoftvert is
a Szoftver ABC-ből!

INFORMÁCIÓKÉRÉS: 51 ▲

DATAEASE
INTERNATIONAL

A DataEase adatbázis-kezelő neve széles körben ismert Magyarországon is. Sőt létezik a teljes egészében magyarul beszélő változata is. Tudja-e Ön, hogy mit jelent ez a név, és hogyan kell kiejteni?

DataEase = détaíz (könnyed adatok). Az ease = íz (könnyedség) szó nem keverendő össze az easy = ízi (könnyű) szóval.

A DataEase cég jelenleg a DataEase MS Windows alatt működő változatát (GUI), a DataEase karakter-alapú rendszer (CUI) egérrel is kezelhető változatát, az SQL front-endként használható DataEase SQL Connect szoftvercsomagot készíti, tehát igen aktívan tevékenykedik.

Ha már a *fejlesztésekben részt venni nem tudunk*, mi is tehetünk valamit e sikeres termékcsalád érdekében. Alkossunk egy magyarul is jól csengő nevet a cég termékeinek!

Kérjük kedves olvasóinkat, *segítsenek* nekünk ebben. És hogy a segítségre egy kissé lelkesítsük is Önöket, a legsikeresebb nevek alkotóinak felajánlunk egy-egy teljes értékű DataEase rendszert. (Ha valaki már korábban beküldött névvel jelentkezik, sajnos nem tudjuk elfogadni jelentkezését, tehát aki hamarabb jelentkezik, az nyer!)

VT-SOFT®
SOFTWARE KFT

A DataEase disztribútora
1033 Budapest, Vörösvári út 103–105.
Tel.: 180-3744 Fax: 180-3750

INFORMÁCIÓKÉRÉS: 55 ▲

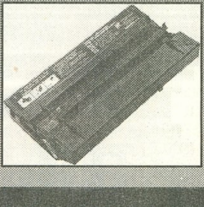
TONER KFT

Másológép és lézerprinter-kazetták felújítása német technológia alapján, garanciával, feketében és színesben.

Europa International

NE DOBJA EL!

Felújítható kazettatípusok: FC, PC, EP, EP-S, EP-L, SHARP Z-30, Z-50.



1095 Budapest Mester u. 21. Tel.: 113-1687 Tel./Fax: 134-3516

INFORMÁCIÓKÉRÉS: 53 ▲

Az IBM „felülemelkedett” a problémák zömén

Az OS/2 2.0 teszi a dolgát

Az OS/2 „él és mozog”, de az igazi kérdés, hogy az új 2.0 verzió elég erőteljes-e ahhoz, hogy elkápráztassa a boldog DOS- és Windows-felhasználókat. Az OS/2-t tekintve a válasz mindig talányos volt.

Az előző verziókhoz képest az új verzió gyors, olcsó és hozzáférhető. Képes a Windows programok futtatására — ráadásul elég egyszerűen —, és „legyőzhetetlen”, ha a DOS programok multitaszkolásáról van szó. Továbbá: a 386-os specifikus tervezés a 2.0-t a legfejlettebb PC-s operációs rendszerré teszi a piacon.

Mindezek azt jelentik-e, hogy érdemes lecserélni a DOS-t, illetve a DOS-Windows kombinációt az OS/2-vel? Mielőtt megtesszük ezt a döntő lépést, ne feledjük, hogy az operációs rendszer cseréje nemcsak az upgrade árát jelenti. Az igazi költség rendszerünk újraépítése és az „átképzés”. Ezek a problémák és az előző verziók gyermekbetegségei tartották vissza a felhasználókat, hogy az OS/2-t univerzális DOS-helyettesítővé tegyék.

A 386SX-es vagy a még fejlettebb rendszerek használói nem hagyhatják figyelmen kívül az OS/2-t. A 2.0 kitelјesítette a 386-os és a 486-os 32 bites lehetőségeit, így a programok gyorsabbak, jobb memória-hozzáférésűek, és kevesebb a rendszerösszeomlás.

Újrakezdés

Az előző verziókhoz képest — amelyek DOS-kompabilitása nem volt az igazi, kevesebb nyomtatót és egyéb perifériát kezeltek — az OS/2 2.0 egy tucat Windows és DOS programot képes párhuzamosan futtatni különböző ablakokban. Lehetőség van több száz fajta nyomtató kezelésére, sőt még DOS-drivereket is futtathatunk. A program teljesíti az IBM célját, azaz „integrált rendszer az OS/2, DOS és Windows programok futtatásához”.

A 2.0-ás az Adobe Type Manager egy teljes változatát is magában foglalja, emellett a miniprogramok és játékok széles skáláját is kínálja.

Az OS/2 legnagyobb erőssége az eredetiség: az elavult DOS foltozgatása helyett az IBM egy teljesen új rendszert tervezett az alapoktól kezdve. Ez azt is jelenti, hogy az OS/2 olyan előnyöket is kihasznál a 386-os és egyéb rendszerek adottságaiból, amelyeket a DOS még „érinteni” sem tud. Ennek következtében falja a rendszerforrásokat. Egy-egy másolatot építettek be a DOS 5.0-ból és a Windowsból — a programok futtatására —, ezért a rendszer elég sok memóriát és merevlemezhelyet foglal el. Bár az OS/2 4 Mbájtos konfiguráción is fut, valójában azonban 6 Mbájt memória és 28 Mbájt szabad lemezterület kell az összes lehetőség kiaknázásához.

Ha elhatároztuk, hogy kell nekünk az OS/2, háromféleképpen installálhatjuk.

A legegyszerűbb, ha csak a C: drive partíciójához adjuk hozzá. Ez kicseréli a DOS-t az OS/2-vel mint elsődleges operációs rendszerrel, ami viszont lehetővé teszi az összes DOS és Windows program átmozgatását az OS/2 Presentation Manager Workplace Shellbe, és a későbbi futtatásukat. A második mód az installálásra, ha az OS/2-t a már létező DOS-partícióhoz adjuk. Ekkor minden bekapcsolásnál választhatunk a két operációs rendszer közül.

Mindkét fenti mód legalább 28 MB szabad lemezkapacitást igényel. Ha nincs ennyi, vagy ha egy harmadikfajta operációs rendszert is akarunk magunknak, akkor használni kell az OS/2 felújított FDISK programját az újrapartícionáláshoz. Ez annyit jelent, hogy többszörös elsődleges és logikai partícionálás lehetséges. Egy lehetőség, hogy 1 Mbájtot adunk a speciális új Multiple Operating Systemhez, két elsődleges partíciót rendelünk az ezt igénylő operációs rendszerhez (DOS, OS/2), valamint egy kiterjesztett partíciót készítünk, amely — OS/2-höz vagy más rendszerhez — többszörös logikai drive-ot foglalhat magában.

A harmadik eset, hogy elmentjük a rendszert, és egy új felosztást csinálunk. Ez kb. egy órát vesz igénybe.

32 bites előnyök

A robbanásszerűen fejlesztett memóriakezelés, és az, hogy a saját eljárásai jól „kommunikálnak”, az OS/2-t a legerősebb 32 bites programmá teszik. A legfontosabb ok, hogy a 32 bites alkalmazások felsőbbrendűek a 16 bites DOS- és Windows-alkalmazásokhoz képest, az ún. flat memory model. A mai 16 bites programokat a 8088-as alapú processzorokra írták, amelyek csak 64 kbájtos — a legnagyobb 16 biten kifejezhető számnak megfelelő — szegmenseket képesek kezelni. Ha nagyobb fájlra van szükségünk, vagy átnyúlik a program a szegmenshatárokon, a DOS és Windows külön tartja nyilván a darabokat, és darabonként rakja össze őket. Ez nagyon lelassítja a működést.

A 32 bites programok akkora memóriát címeznek meg, amekkorára szükségük van, egészen 4 Gbájtig. Az összeillesztési időket megtakarítva a 32 bites programok gyorsabban futhatnak.

Az OS/2 természetesen képes a 16 bites programok futtatására is — melyek OS/2 1.x, DOS, Windows alá készültek —, és ezek azért gyorsabbak az OS/2 alatt, mert kihasználják a párhuzamos futást. Az Aldus Pagemaker for OS/2 PM egy példa erre. Ennek a programnak a sebessége ugyanis nagyban azon múlik, hogy a párhuzamosítható eljárásokra — átméretezés, betöltés, nyomtatás stb. — képes-e a program, illetőleg az operációs rendszer.

Egy érdekes technikai kérdés, hogy az OS/2 hogy áll a valódi multitaszkolással: a program képes-e biztonságosan és folyamatosan lefutni a háttérben — mindenféle speciális programozási eljárás nélkül. Az operációs rendszer határozza meg, melyik program foglalja le a processzort, szabadítja fel azt más alkalmazások számára, illetve határozza meg a beosztást a programok között.

Ha a valós különbségeket akarjuk meghatározni a DOS-alapú Windows

és Desqview, valamint a valódi multitaszkolás között, elég nehéz dolgunk van. (Csak az IBM, a Microsoft és a Quarterdeck tudja igazán ezeket.) A két „filozófia" közötti választáshoz ismernünk kell a következő alapvető különbségeket.

A valódi 8086-os

Először nézzük a DOS-t, mert ezt a 8088-as, 8086-os processzorokra tervezték — egy felhasználó, aki egyszerre csak egy feladattal foglalkozik. Ekkor ugyanis kevesen gondolták, hogy a felhasználók több programot akarnak egy időben futtatni. Ennek következtében a chipek szempontjából is azt feltételezték, hogy az aktív program használja az összes erőforrást.

Évekkel később a programgyártók elkezdték a taszkkapcsoló programok készítését, ahol nem kellett kilépni az egyik programból ahhoz, hogy egy másikba lépjünk.

A multitaszkos kinézetet úgy valósították meg, hogy az egyik program „aludt", míg a másikon dolgoztunk. Az IBM Topview, MS Windows és a Quarterdeck Desqview voltak az elsők ebben kategóriában. A 386-os kibocsátásáig kellett arra várni, hogy a multitaszkolás valósággá válhassék.

Látszólagos 8086-os(ok)

A 386-os chip áttörést jelentett a multitasking terén, ugyanis ez volt az első processzor, amelyik a Virtual 8086 módot tudja. Ebben a módban a processzor létrehoz több látszólagos DOS-gépet (VDM, azaz virtual DOS machine) — ezek olyan memóriaterületek, mintha egyedülálló PC-kkel rendelkeznénk.

Minden VDM-nek megvan a saját kiviteli és periferiális forrása, valamint a saját 640 K RAM-ja, mínusz a DOS és a driverek által elfoglalt terület. A VDM-eken futó programoknak nem kell megosztaniuk semmit a többiekkel.

A Desqview 386 volt az első program, amelyik kihasználta a Virtual 8086 módot a valódi multitaszkolásra. A következő a Windows volt, a 386 Enhanced mód megvalósításával. A Windows képes vegyes DOS- és Windows-alkalmazások multitaszkolására, de ez mégsem valódi, ugyanis minden Windows és DOS program ugyanabban a VDM-ben fut, azaz nincs kihasználva a Virtual 8086 mód.

Valójában a Windows azt valósította meg, hogy az egyes programok időszeleteket használnak a processzoridőből, majd visszaadják a processzort a főalkalmazásnak. Van, akinek elegendő pusztán az ilyen multitasking.

Sok felhasználó azonban igényli a valódi lehetőségeket. Speciálisan igaz ez hálózatokban vagy akkor, ha bonyolult műveleteket — például táblázat újraszámolása, programfordítás, dokumentum formázása stb. — akarunk a háttérben végeztetni. Az OS/2 mentes azoktól a behatárolásoktól, amelyeket a Windowsnál tapasztalhatunk. Ez a program a DOS- és Windows-alkalmazásokat a VDM-eken valódi multitaszkolással futtatja. Egyetlen fogyatékosság, hogy nem minden korábbi — speciálisan a korábbi OS/2-re tervezett — program fut rajta.

DOS- és Windows-boxok

A memóriakezelés és a rugalmasság: e két előny teszi a Windows 386 Enhanced módnál alkalmasabbá az OS/2 2.0-t a DOS programok multitaszkolására. Az OS/2 kétfajta virtuális gépet ajánl a DOS-alkalmazások futtatására: ún. külső és belső boxokat.

A belső DOS-box egyszerűen emulálja a DOS-t, azaz minden DOS programhoz 634 K-t használhatunk. Nem használjuk fel a memóriát az egérhez vagy a hálózati modulokhoz, ezeket ugyanis a rendszer a boxon kívül helyezi el, ami lényeges különbség a Windowshoz képest. A virtuális boxok megcserélhetőek, emiatt aztán annyi DOS programot futtathatunk, amennyit a memória és lemezkapacitás lehetővé tesz.

A külső DOS-box megengedi, hogy betöltsük a DOS bármelyik korábbi verzióját. A multitasking során futtathatunk olyan programot, amely a DR DOS-t igényli, de ezzel egy időben más helye(ke)n a DOS 5.0-t vagy egyéb verziót használjuk. Mivel minden DOS-box a saját DOS-verzióját tölti be, ezért csak 590 kbájtnyi szabad memória marad.

Az OS/2 tartalmazza a Windows 3.0-t (a 3.1 később kerül majd bele), amellyel teljes képernyős Windows-programok multitaszkolása is megvalósítható. Minden Windows program a saját boxában fut, ezáltal az adatvédelem még a Windowsén is túltesz.

Az a képesség, hogy adatokat cserélhetünk a DOS-, Windows- és OS/2-alkalmazások között, egy másik extra szolgáltatás. Az IBM módosította a clipboardot a Windows programban, így tetszőleges módon vághatunk ki, illetve illeszthetünk be adatokat a Windows- és OS/2-alkalmazásokból (-ba). A végleges 2.0 verzió alkalmas arra is, hogy egymás melletti, átméretezhető ablakokban fussanak vegyesen a Windows-, DOS- és OS/2-alkalmazások.

A grafikus kernel 32 bitesre szabása még várat magára, de a következő hónapokban ez is megtörténik, s az új grafika még jobban felgyorsítja majd a rendszert.

Többszörös fájlrendszerek

A 2.0 további lényeges újdonsága, hogy támogatja a többszörös fájlrendszert. A DOS-ban és Windowsban a FAT-tábla használata eléggé lelassította a fájlműveleteket. Az OS/2-ben lehetőség van a régi FAT-tábla lecserélése az ún. High Performance File Systemre (HPFS-re); ez úgy tárolja a fájlok információit, hogy a nyitás, zárás, hozzáférés, elmentés sokkal gyorsabbá válik.

A HPFS megszünteti a FAT miatti, maximum 11 betűs fájlneveket: most már 254 betű hosszúságban adhatunk nevet az állományoknak. Az IBM Workplace Shell azonban azt is lehetővé teszi, hogy a hagyományos FAT-tábla használatakor is használhassunk 254 karakteres neveket. A HPFS csapdája, hogy a vele használt partíciót újra kell formázni.

Utolsó gondolatok

Az IBM kijavította az OS/2 két fő hiányosságát — a kevés driverről és a DOS programok behatárolt támogatásáról van szó —, amelyek az 1987-es bemutatás óta késleltették az áttörést. Azt viszont, hogy a program képes lesz-e végre elfoglalni megérdemelt pozícióját, a rendszer hosszú távú stabilitásától függ, valamint attól, hogy mennyire lesz képes az összes Windows program futtatására.

Biber Attila

Barex ÚJ!

COMPUTER

SZÁMÍTÁSTECHNIKAI SZAKÜZLET

KÉT ŐSZI ÚJDONSÁG:

FOTOBASE

GRAFIKUS ADATBÁZIS-KEZELŐ SZOFTVER

Színes adatbázis-kezelő,
képekhez szöveges információ rendelhető,
keresési funkciók. Hivatkozás, csoportképzés.

IRIS 1.1

SZÁMLÁZÓ ÉS RAKTÁRKEZELŐ PROGRAM GRAFIKUS MEGJELENÍTÉSSEL

Számlázás, szállítólevél, árajánlat, bevételezés, statisztikák, zárások. Kétmonitoros rendszerrel a színes monitoron a termék grafikus képe megjelenik.

Kérjen demóverziót!

BAREX COMPUTER
1054 Bp. V., BAJCSY-ZSILINSZKY ÚT 54.
TELEFON: 111-6025, 131-0946

INFORMÁCIÓKÉRÉS: 04 ▼

Déma
Számítástechnikai Kft.

1092 Budapest IX., Ráday u. 47. • Tel./Fax: 117-1251

AT 386/33 MHz számítógép (2 MB RAM, 1,2 MB FDD, 80 MB HDD, 2-szer soros-párhuzamos kártya, 14" monokróm monitor, 101 gombos billentyűzet)	83 000,–
80 MB winchester (WD286)	23 900,–
210 MB winchester (WD)	48 900,–
AT 386/33 MHz, 64 kB cache alaplap	19 000,–
14" SVGA monitor	25 900,–

Áraink az ÁFA-t nem tartalmazzák!

INFORMÁCIÓKÉRÉS: 17 ▼



INFORMÁCIÓKÉRÉS: 33 ▼

A „titkos" tudás

Források és kútfők...

A tudás begyűjtésének már kialakult egy, a gyakorlati esetek többségében jól használható módszer-, illetve eszközcsoportja; ezeket (is) szándékozik ismertetni a következő néhány hónapban rovatunk. Most — áttekintésként — mielőtt az egyes konkrét eljárásokra sort kerítünk, célszerű kitérni magára a tudásbeszerzési folyamatra, hogy jobban érthető legyen a felmerülő problémák természete.

A szakértő rendszerek — vagy a ma már egyre terjedőbb nevükön tudásalapú rendszerek (TR-ek) — fejlesztői között meglehetős az egyetértés abban, hogy az ilyen alkalmazások építésének legkritikusabb pontja a releváns (szak)tudás beszerzése, és számítógépen implementálható formába öntése. Mivel e rendszerek teljesítőképessége (mint ahogy az emberi szakértőé is) alapvetően a problémamegoldáshoz mobilizálható szaktudás mennyiségétől és minőségétől függ, az itt mutatkozó nehézségek lassíthatják (és lassítják is) a TR technika elterjedésének ütemét. Mivel pedig ez ma már nemcsak a szakterület belső fejlődését lassítja, hanem komoly üzleti érdekeket is érint, jelentős kutatói kapacitás foglalkozik a módszerek javításával.

Magának a tudás beszerzésének folyamata azt célozza, hogy az emberi szakértő által birtokolt, a problémamegoldásban eredménnyel kecsegtető tudást és annak felhasználási módszereit a szakértőből kinyerjük, és a számítógép által „emészthető" alakra hozzuk (formalizáljuk). Ez ma még nem automatizálható teljesen: a szakértőn és a számítógépén kívül egy harmadik közreműködőre is szükség van — ő a tudásmérnök. A másik kettő viszonylatában egyfajta zsilip: feladata lényegében az, hogy a gyengén formalizált szakértői tudás szemantikai gazdagságából a lehető legtöbbet „emelje át" egy formális (számítógépi) reprezentációba — a tudásbázisba.

A szakértő nem az egyetlen tudásforrás: a tudásmérnök felhasználhatja a szakterület írott anyagait, esettanulmányokat, cikkeket stb. A korszerű hipermédia technika nagy segítséget nyújt az ilyen anyagok strukturálásában és elemzésében.

A tudásbeszerzési folyamat szereplői tehát a szakértő, a tudásmérnök és a számítógép. Tapasztalati tény, hogy a tudásbázis (a számítógépre vitt tudás) torzítva tartalmazza a szakértő fejében „élő, valódi" ismereteket: a fenti három szereplő különböző módon befolyásolja a végeredményt.

A „tudatalatti" faktor

Ami a szakértőt illeti, a leggyakoribb nehézséget az okozza, hogy azok a tudáselemek, amelyek a hatékony problámamegoldásban a legfontosabb szerepet játsszák, nehezen vagy egyáltalán nem verbalizálhatók (ezt nevezzük rejtett tudásnak). Ide tartoznak mindazon, a szakmai praxis során automatikussá vált problémakezelési technikák, amelyek az egyes részfeladatok megoldását kivonják a tudatos tevékenység köréből, lehetővé téve ezáltal a szakértő számára a feladat lényegi részeire, specifikumaira történő fókuszálást.

Ezeket a tudáselemeket „procedurális tudás" néven ismeri az irodalom. Pontosan ezek azok a részei a szakértői tudásnak, amelyek — miközben mind hatékonyabbá és autonómabbá válnak, egyre jobban eltűnnek a tudatos elérhetőség köréből: gyakorlatilag a szakértő nem tud számot adni róluk. Ezenkívül persze még számos pszichológiai tényező befolyásolja, hogy a szakértőből milyen mennyiségű és minőségű tudásanyag nyerhető ki. Ilyen a nem akaratlagos tudásvisszatartás, ami például olyan esetekben következik be, amikor a „donor" saját pozícióját véli veszélyeztetni az elkészítendő rendszertől. (Gyakorló tudásmérnök számára nagyon megszívlelendő szempont!)

A másik torzító tényező a tudásbeszerzési folyamatban maga a tudásmérnök, akinek a hatékony munkához feltétlenül rendelkeznie kell bizonyos ismeretekkel arról a területről, amelyre a szakértő rendszert fejleszti. Ugyanakkor ő maga általában informatikai alapképzettségű szakember, a mesterséges intelligencia specialistája, aki a szakértő által mondottakat a saját „kognitív szűrőjén" keresztül interpretálja (anélkül, hogy ennek különösebben tudatában lenne). Ez azzal az egyenes következménnyel jár, hogy az eredeti tudásanyaghoz képest az interpretált változatban hangsúlyeltolódások, idegen szempontok (a tudásmérnök preferenciái) jelenhetnek meg. Végeredményben tehát a szakértő—tudásmérnök interakciók során előálló, ún. „tudásbázis-nyersanyagban" egyfelől hiányozhatnak a szakértőben eredetileg még meglévő tudáselemek, másfelől pedig a szakterülettől idegen szempontok, preferenciák kerülhetnek bele.

Fontos ugyanakkor megjegyezni, hogy szemben a teljesen automatikus technikákkal, a szakértő—tudásmérnök interakción alapuló tudásbeszerzésnek van egy nagy előnye, ami az ember—ember közötti kommunikáció kontextusérzékenységével függ össze. Arról van szó nevezetesen, hogy a tudásmérnök olyan információkat is ki tud emelni a szakértővel történt párbeszédből, amelyek nem kerültek explicite megfogalmazásra, de beleesnek az „ezt mindenki tudja" (csak a számítógép nem), vagy a „hiszen ez az eddigiekből világos" kategóriába. Tipikusan ilyenek az emberek hétköznapi tudásanyagának az elemei (a tárgyak lefelé esnek, a férfiak nem lesznek terhesek stb.).

„Táplálék" — kissé nyersen

A tudásbázis-nyersanyag még nem tudásbázis, ahhoz túlságosan lazán szervezett, nem alkalmas még közvetlenül a számítógépen történő implementálás-

ra (legtöbbször interjúprotokollok, szövegdarabok, video- és grafikus anyagok többé-kevésbé rendezett halmazából áll).

A következő lépésben ezt az anyagot „gépre kell rendezni", a tudást formalizálni kell: ez ismét a tudásmérnök feladata.

Ma már többféle tudásreprezentációs technika nagyszámú implementációja áll a tudásmérnök rendelkezésére: a mesterséges intelligenciában használatos nyelvektől az igen kiérlelt és flexibilis szakértői keretrendszerekig. Természetesen mindezen eszközök a létező összes hardver- és szoftverkörnyezetben hozzáférhetők (nagyon jó keretrendszerek kaphatóak már 500 dollár alatti áron is, például a Level5 object). A tudásmérnök dolga abban áll, hogy megtalálja a „gépesítendő" tudáshoz legjobban illeszkedő tudásreprezentációs technikát, és az ezt megvalósító eszközt.

Annak a folyamatnak a során, amikor a tudásmérnök a tudásbázis-nyersanyagot egy keretrendszer vagy nyelv által adott formális leírásba viszi át, újabb „tudásvesztés" következhet be a reprezentáció korlátai miatt. Az ún. hibrid keretrendszerek többféle tudástípus ábrázolását is lehetővé teszik, de még nincs olyan eszközünk, amellyel a szakértői tudás minden elemét implementálni lehetne.

A források „befoglalása"

A továbbiakban (szinte csak címszavakban) vázoljuk, milyen módszerek állnak a tudásmérnök rendelkezésére ahhoz, hogy a szakértői tudást a felszínre hozza. A két fő csoport az automatikus (gépi) tanulásé, illetve az interaktív módszereké.

Az interaktív módszerek ismét két csoportra bonthatók: a direkt és az indirekt eljárásokra.

Direkt technikák: interjú, protokollelemzés, kontextusszűkítés, megszakításos elemzés, közvetlen megfigyelés.

Indirekt technikák: többdimenziós skálázás, az RGA módszer, a hierarchikus clusterek módszere, a rendezett memóriafelidézési fák módszere.

Anélkül, hogy a részletekbe belemennénk, a két csoport közötti legfontosabb különbséget előre megfogalmazhatjuk. Míg a direkt technikák mindegyikénél a szakértő és a tudásmérnök közötti (géppel támogatott vagy anélküli) interakció a domináns elem, addig az indirekt módszerek esetében a tudás kinyerésének döntő lépései a szakértő és (egy pszichológiai megfontolások alapján kialakított) program között zajlanak.

Természetesen mindkét csoportnak megvannak a maga előnyei és hátrányai, alkalmazhatósági követelményei: más-más tudástípusok kinyerésére általában különböző technikák válnak be. Például az RGA módszer igen jól használható a szakértő tudásában jelen levő preferenciarendszer feltárására, míg a hierarchikus clusterek módszerének a segítségével a tudás taxonomikus szerveződésére nézve kaphatunk információkat.

Ezek a tudáselemek általában a korábban említett „rejtett tudás" kategóriájába tartoznak, így az indirekt technikák a személyesebb, viszonylagosabb transzfer mögött rejtve maradó elemeket hozzák felszínre. Ezzel szemben a direkt módszerek a valamilyen „megnyilvánulással" kísért szakértői tevékenység rögzítésére váltak alkalmasabbakká.

Megjegyzendő, hogy a direkt technikák, noha általában szemantikailag gazdagabb végeredményt produkálnak, érzékenyebbek mind a szakértő, mind pedig a tudásmérnök oldaláról kiinduló torzító hatásokra.

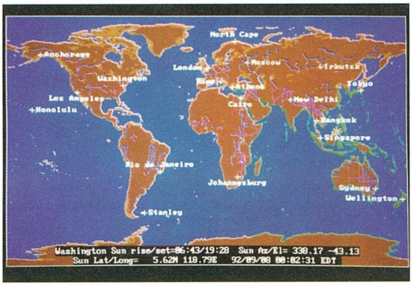
Koch Péter

Világóra — világtérkép

A Skyglobe Star Glazer program elsöprő sikere valószínűsíti, hogy az #588 számú SolarSoft-lemezen található két földrajzi-csillagászati program is széles körben válik népszerűvé.

A Geoclock nevű világóra gyönyörű grafikájú térképen mutatja az egyes városok helyi idejét. A kiszámított és megjelenített időpontok számítógépünk belső órájának felelnek meg. Számos paraméter beállításával variálhatjuk a program lehetőségeit. Bemutathajuk a Nap pillanatnyi helyzetét, a Föld részeit napfényben és szürkületkor, a helyi napfelkelte-naplemente időpontját... A térkép aktualizálása automatikusan történik néhány másodpercenként. A 16 színű EGA/VGA grafikából a maximumot hozták ki a fejlesztők.

A Globedrawer programmal tetszőleges szögből szemlélhetjük bolygónkat. A dokumentáció tévesen CGA-grafikára is alkalmasnak tartja, igazából csak Hercules, EGA, VGA esetén működik problémamentesen. Automatikusan felismeri rendszerünk grafikus kártyáját.



Szöveges állományban tárolja a program az általa ismert helységek neveit, hosszúsági és szélességi koordinátáikkal együtt. (Nekünk itt a szélesség a fontos!) Az egyes nevek 25 karakteresek lehetnek.

A program indításakor néhány paraméter beállításával elérhetjük, hogy a megjelenítés a számunkra legkedvezőbb legyen. Elsősorban a három szög megadására kell ügyelnünk, ez határozza meg ugyanis a Föld helyzetét a képernyőn. (Fok, perc, másodpercben kell megadnunk a szögeket.) Az áttetszőség is három fokozat között változtatható, a rácsozat beosztását is módosíthatjuk. Megadhatunk olyan helységnevet is, amely nem szerepelt a program nyilvántartásában. (Auto-módban egymás után megjelenik a program által ismert összes hely.)

Szőke Péter

Őszi újdonságok a SolarSoftban

Lemezszám	Programnév
572	Kung Fu Loui (3 lemez)
573	Hugo's House of Horrors
574	Hugo II - Whodonit
575	Hugo III - Jungle of Doom (2 lemez)
576	Dark Ages
577	The Last Half of Darkness
578	Image Alchemy (2 lemez)
579	AlcShell
580	Mercury
581	Offix
582	Chemie-Star
583	Q&A Study Aid
584	Title Maker
585	ProWindows LITE
586	3D-LIFE
587	Nervous System Const. Kit
588	Geoclock; Globe

Lemezszám	Programnév
589	Moraff's World
590	Unnkulion Underworld
591	Commander Keen I.
592	Commander Keen IV. EGA/VGA (2 lemez)
593	TEGL Klondike
594	TEGL MahJongg Solitair
595	Duke Nukem



Humorérzék: kötelező!

A SolarSoft programkönyvtár egyik népszerű programja a „szerelmes DOS" (LoveDOS — #344 számú lemez.) Ennek sikerén felbuzdulva kerültek a katalógusba azok az új programok, amelyek alkalmasak arra, hogy némi derűt csempésszenek a munkás hétköznapokba. De vigyázat! Csakis jó humorérzékkel megáldott környezetben számíthatunk a sikerre! Nem mindenki vevő ezekre a gegekre.

Vegyük tehát sorra! Mike van Pelt Slod nevű TSR programja a háttérben várakozik arra, hogy valaki az ALT-CTRL-DEL billentyűkombinációval újraindítsa a gépet. Ekkor egészen drámai angolsággal panaszkodni kezd a gonosz felhasználóra. A hatást tovább fokozhatjuk, ugyanis a szerző mellékelte az assembler forráskódot. Ebben tetszés szerint átalakíthatjuk az angol siránokat ékes magyar nyelvre.

Ismeretlen szerző alkotása az April nevű program. Azt lesi, hogy leütünk-e egy billentyűt, ekkor ugyanis fejre állítja a képernyőt. A következő billentyű leütésekor pedig visszaáll az eredeti helyzet. Ez hamarosan az őrületbe kergetne bárkit, de szerencsére az ESC-pel leállíthatjuk a körforgást.

A Monster is alkalmas eszköz kollégáink riogatására. Amikor gyanútlanul elindítják egy programjukat, eleinte semmi sem történik. Az M billentyű leütésekor azonban megjelenik a képernyőn egy szörnyeteg (M — Monster — magyarul szörny), BOOH felirattal a szájában. Alan Meiss egy másik hasonló programot is készített, BOO! néven. Itt nem ragaszkodik az M billentyűhöz, bármely billentyű leütésekor látjuk a szörnyet, gépünk hangszórójából pedig a BOOH-hoz hasonló hangot hallunk. Úgy tűnik, hogy Alan Meiss fantáziáját megragadták a számítógépes tréfalehetőségek. Replace nevű memóriarezidens programja ugyanis a képernyőn talált szöveg egyes szavait véletlenszerűen lecseréli más szövegből vett szavakra. A megtréfált szavak hossza 2 és 8 karakter közötti, mindig ugyanolyan hosszú az új szó, mint amit helyettesít. Így a szöveg nem fog eltolódni. A program indításakor megadhatjuk annak a szöveges állománynak a nevét, ahonnan a programnak a helyettesítő új szavakat vennie kell. Bárki szabadon eresztheti fantáziáját. Képzeljük el kollégáink meglepetését, amikor például egy hivatalos levélben ahelyett, hogy „Kelt, mint fent", lakonikusan ennyi áll: „Kösz, mára elég".

A Windows-kedvelők tábora sem marad ki a „jóból". Scott Gourley Icofritéjában például az ikonok egyenesen menekülnek az egérkurzor elől! Nem mindennapi élmény kollégánk arcát figyelni, miközben elszántan üldözi ezeket a menekülő ikonokat! Még ha sikerül is az ikonokat a képernyő szélén sarokba szorítania, korai az öröm. Ekkor ugyanis kilép a képből, eltűnik, majd pedig a képernyő egy teljesen váratlan helyén újra feltűnik. A program működését bizonyos mértékig módosíthatjuk is. Erre szolgál egy egyszerű kis menü, amelyet az ALT-Space kombináció aktivizál. Némi gyakorlással el is tüntethetünk programikonokat. Ugyanebből a menüből ki is kapcsolhatjuk a programot, hiszen a legjobb viccből is elegünk lesz idővel.

Randy L. Eccles Barteyes II nevű, Windows-környezetben futó programjában az ismert rajzfilmfigura, Bart Simpson szemei töltik be a képernyőt, és követik az egérkurzort. A program indításakor és befejezésekor némi gépi hang is hallható. Ez a „Hi Man" angol üdvözlés nem túl jól sikerült imitációja.

Egy jó tanács: bármelyik tréfás időzített bombával akarjuk is meglepni gyanútlan ismerőseinket, maradjunk mindenképpen kiszemelt áldozataink közvetlen közelében! Nagyon rosszul sülhet el, ha éppen álmos, kedvetlen kolléga humorérzékére apellálunk.

Verebély Pálné

INFORMÁCIÓKÉRÉS: 24 ▶

Printerportok csereberéje — egy helyben ülve

A lustaság fél egészség!

Aki lusta, az valószínűleg ráérős.
Aki gazdag, annak minden bizonnyal
négy printere is akad...
Akinek kettő van, feltehetőleg nem egészen lusta,
de nem is nagyon tehetős.
Ha kicsit ráér, feltétlenül olvassa el ezt a cikket!

Mire jó ez a kis program, amelyről most szó lesz? Ki veheti hasznát? Mindazok, akik restek felállni, és mondjuk az LPT2-ről nem akarják áttenni a printerkábelt az LPT1-re.

No de egyáltalán miért kell ez?

Ha véletlenül nem jó helyre dugtuk volna a printerport csatlakozóját, és nem akar a nyomtatónk nyomtatni, akkor a program elindítása után — felcserélve a két portcímet — már valószínűleg hajlandó lesz a nyomtatónk működni.

Vagy: ha a számítógépünkhöz két nyomtató van csatlakoztatva — egy egyszerűbb, olcsóbb, de gyengébb minőségű, és egy drágább, de jobb minőségű, mondjuk egy lézernyomtató —, akkor a kinyomtatandó szöveget először célszerűbb a gyengébb minőségű nyomtatón kinyomtatni. Az esetleges hibákat így is kijavíthatjuk, majd amikor már a szöveg hibátlan, csak akkor célszerű a lézerre vinni. (Ugyanez a cserebere irányulhat a COM portokra is: ha egyetlen címet módosítunk a rutinokban, akkor azok a COM1-et fogják felcserélni a COM2-vel.)

Tudnivaló ugyanis, hogy az IBM PC-k a bekapcsoláskor, a bootoláskor helyezik el az LPT1, LPT2, LPT3, LPT4 címeit a BIOS adatterületén; a 40:0h, 40:2h, 40:4h, 40:6h címeken a négy lehetséges printerport címeit — és ehhez teljesen hasonlóan a COM1, COM2, COM3, COM4 címeket pedig a 40:8h-tól kezdődően, és kétbájtos lépésekben, tehát a 40:0Ah, 40:0Ch, 40:0Eh címeken (a négysoros kommunikációs port címeit). Ezeket a címeket írhattuk volna természetesen másként is: 0:400h, 0: 402h stb.

Mivel ez RAM-terület, a címek megváltoztathatók, módosíthatók. Sokan nyilván korábban is, mi több, gyakorta megváltoztatták ezeket, ha szükség volt rá. Leginkább a DOS-hoz tartozó DEBUG segítségével, amely tetszőleges memóriacím módosítását, felülírását teszi lehetővé, de amelynek használata a számítógépet kezelő emberek tekintélyes hányadának mégsem egészen kézenfekvő. Főleg amiatt, mivel itt a címeket is és a memóriatartalmakat is a hexadecimális számrendszerben kell megadni. Ez pedig a nem gépi kódban programozók számára gyakorta kényelmetlenséget jelent.

A rutinok igen rövidek, 50 bájtnál kevesebb helyet igényelnek. Az első program, a CSERELPT.ASM lehetne egy kicsit tömörebb is, ami egy kicsit gyorsabb működést is eredményezne. Így keletkeztek a variációk: a CSERELP2.ASM, CSERELP3.ASM, CSERELP4.ASM. El szabad tűnődni azon, hogy mégis vajon melyik a legtömörebb, illetve a leggyorsabb (figyelmen kívül hagyva a kód elején, illetve a végén lévő, a felhasznált regisztereket elmentő, illetve visszaállító kódrészleteket).

Mindegyik itteni ASM programra vonatkozik, hogy először a MASM segítségével kell lefordítani őket objekt modullá (.OBJ), majd a LINK segítségével lehet őket futtatható programmá tenni (.EXE), amlyet aztán az EXE2BIN program felhasználásával lehet az igen rövid és tömör COM fájllá átalakítani.

Akinek nem elég a két első port felcserélése, hanem mind a négy lehetséges portot variálni szeretné, az a RING_LPT.ASM segítségével ciklikusan körbe tudja forgatni a négy nyomtatóport címeit. Erre valószínűleg igen kevés embernek van szüksége, mert a legritkább esetben van egyszerre négy nyomtató a számítógéphez kapcsolva. (Párhuzamos és soros kimeneteket előállító kártyák is kellenek ehhez — a nyomtatókon kívül.) Biztos, ami biztos: érdemes a már említett DEBUG program segítségével megnézni, hogy mind a négy LPT vagy COM port címe megtalálható-e a BIOS adatterületén.

Ehhez a következőket kell tennünk. Indítsuk el a DEBUG-ot, majd pedig adjuk ki a következő utasítást:

-d 40:0 L 10

Ha az első 8 bájton nullától eltérő számokat találunk, akkor mind a négy printerportunk létezik. Ha a következő 8 bájton is ugyanez a helyzet, akkor a négy darab soros COM portunk is megvan. Egy eredeti IBM PS/2 esetén például a helyzet a táblázat szerinti.

Ez tehát azt is jelenti egyúttal, hogy itt az első printerport címe 3F8 hexadecimális formában, amely megfelel a decimális (3 * 256) + (15 * 16) + (8) = 1016 -nak.

Ha már nem kívánjuk tovább nézegetni a bájtokat, akkor a

-q

hatására léphetünk ki a DEBUG programból. Amennyiben mind a négy LPT port létezett, akkor érdemes használni a RING_LPT.ASM nevű programot. A COM portok esetében szigorúbb a helyzet. Csak azok fordítsák le és használják a négy COM portot felcserélő programot, akiknek ténylegesen is létezik ennyi COM portja, mert különben meglepetésekben lehet részük, a gépük könnyen lefagyhat.

Egyébként — ha négyszer futtatjuk le a RING-LPT.COM programot — a portcímek újra visszaérkeznek az eredeti helyükre.

Szabó Péter Pál

```
A DEBUG promptja [bejelentkező jele]
↓
– D 40:0 L 10        ⇐  a DEBUG parancsa [ (D)ump: Listázás ]

0040:0000  F8 03  F8 02  20 32  00 00 — BC 03  00 00  00 00  C0 9F  .... 2.........
   ⇓         ⇓      ⇓      ⇓      ⇓       ⇓      ⇓      ⇓      ⇓
Memória    LPT1   LPT2   LPT3   LPT4    COM1   COM2   COM3   COM4
címe
```

Aki beleesett a kútba

**Az előző számban olyan Windows-játékokról írtunk, amelyek élvezetéhez nem kevés logikára és gondolkodásra van szükség.
Most viszont olyan programokat mutatunk be, amelyek a felhőtlen kikapcsolódást szolgálják.**

Úgy tűnik, a Tetris-utánzatok (-utánérzések, -továbbgondolások) végeláthatatlan sorban követik egymást. A Hextrisben a játék nevének megfelelően az elemeket hatszögek alkotják. Maguk a hatszögletű elemek is hatfélék lehetnek, akár a hagyományos Tetrisben.

A darabok forgatásával nem négy, hanem hat állás lehetséges, például a hosszú darabot nem lehet vízszintesre beállítani, csak függőlegesre, vagy úgy, hogy jobbra, vagy úgy, hogy balra dől. Ha egy vízszintes sort megtöltöttünk, eltűnik.

Mivel a „kutat" hatszögekkel töltjük ki, sokszor előfordul, hogy nem vízszintesen, hanem ferdén építkezünk. Ilyenkor persze könnyebb csálén leejteni a darabokat. Segítséget jelent az, hogy a gép mutatja a soron következő darabot. Aki viszont „vakon" kívánna játszani, bosszankodhat, mert a funkciót nem lehet kikapcsolni, és a ponttáblázat is nagyon hiányzik. Így a rekordokat, ha valakinek nagyon fontos, külön kell megírni.

A második játék szintén egy klasszikus — az úgynevezett packman — továbbfejlesztése. Biztosan mindenki látott már olyat, ahol egy labirintusban almákat, golyókat stb. kell megennie egy kígyónak, úgy, hogy ne ütközzön a saját farkába. Nos, a Snakes egy kicsit megcsavarja a szerepeket. Itt egy kis sárga buldózerrel vagyunk, körülöttünk félelmetes és óriási kígyók tekeregnek. De mi halált megvető bátorsággal közéjük merészkedünk, a pályán található kék téglákat rájuk toljuk, és ezzel a hátsó részük eltűnik. Ha sikerül a fejükre tolni a téglát, az egész kígyó megsemmisül.

A feladat persze nehézséget is rejt magában, ha egy kígyó nekünk jön, akkor elveszítünk egy életet. Összesen négy buldózerünk van, a maradékot a képernyő bal felső sarkában láthatjuk. A pályán néha feltűnnek még kis piros „erőkövek", ha ezeket sikerül behabzsolnunk, nem kell kék téglákat tologatnunk, közvetlenül is nekieshetünk a kígyóknak. Ilyenkor sokkal könnyebb pontokat gyűjteni, úgyhogy érdemes üldözni a „fenevadakat". A kígyók nem tudják eltolni a téglákat, de a szürke színűeket mi sem, azok számunkra is akadályok. Ha a kígyót vagy egy részét megsemmisítjük, minden darabjáért 20, a fejéért 50 pontot kapunk. Eredményünk bekerülhet a rekordtáblázatba. Itt egy kicsit vigyázni kell, mert ha nem az egeret használjuk az OK gomb lenyomására, hanem az Entert, figyelmetlenségünkben könnyen kitörölhetjük az egész táblázatot, és ezzel elvész az összes addigi eredményünk. A megoldás: amikor a gép megmutatja a rekordokat, a jobbra vagy balra nyilat nyomjuk meg, és csak utána az Entert.

Érdekes, hogy az egérkurzor helyett „use keys", azaz „használd a gombokat" felirat olvasható. Az egérrel csak a Pause, vagyis Szünet gombot és a Windowsban már megszokott legördülő menürendszert irányíthatjuk. Az utóbbival a hangot és az „erőköveket" kapcsolhatjuk ki/be, új játékot kezdhetünk, vagy megnézhetjük a rekordtáblázatot. Itt lehet a játékból kilépni is. Különben a programhoz tartozik egy elég egyszerű, kezdők által is könnyen érthető Segítség opció, ami ugyanolyan, mint amit a Windows programoknál már megszoktunk.

A harmadik játék a kisgyerekekre gondol, a neve Blackout. Feladata megtanítani az egér pontos kezelését, de kikapcsolódásnak is megfelel.

Amikor elindítjuk a játékot, egy színes téglalapokból álló nagy téglalapot látunk. Érdemes maximálni a táblát, akár a bal felső sarokban lévő menüvel, akár a jobb felső sarokban lévő felfelé mutató háromszöggel. A képernyőn célkörök, színes mezők és számok vannak. Feladatunk a célköröket eltalálni az egérrel, mielőtt az összes színes mező feketévé nem válik. Ezt úgy érhetjük el, hogy ráállunk, és az egér bal gombját megnyomjuk. Ha minden célkört eltaláltunk, új pálya kezdődik, ahol több a cél, és gyorsabban feketül a képernyő.

A játék még további érdekességeket is kínál. Ilyenek a 2X, 3X, ...5X feliratú mezők. Ha ezekre „rákattintunk", a célkör után kapható pontszám kétszeresét, háromszorosát stb. lehet kapni. Vannak még a nyilak, amelyek szinte átsüvítenek a képernyőn. Ha ezeket valami véletlen folytán sikerül eltalálni, az összes addig megfeketedett mező visszaváltozik színessé. Vigyázni kell, mert sikertelen találat esetén a rossz mező és a környező nyolc is befeketedik. Van még egy különleges figura, akit Boogymannek hívnak. Ha megjelenik, csak őt lehet eltalálni, az összes egyéb akciónk sikertelen lesz. Szimpatikus vonása a játéknak, hogy sok dolgot mi magunk is beállíthatunk. Mindhárom játék a SolarSoft programkönyvtár Windows-játékai között kapott helyet.

Gerlits Judit

A TopSpeed programozási nyelvcsalád

„Szem-szájnak ingere"

Szoftver-összeállításunkban ezúttal a TopSpeed compilereket mutatjuk be (C++, Pascal, ANSI C és Modula-2). Ezek jelentősen eltérnek azoktól, amelyeket azelőtt használtunk. Mindegyik pontosan beleillik az ún. TopSpeed fejlesztői környezetbe (TopSpeed Development Environment), mely magában foglal minden elképzelhető programfejlesztői igényt és segédeszközt: optimalizált kódgenerálást, editort, automatikus linkert és függvénykönyvtárazót (librarian), debuggert, projektkezelő rendszert, hipertext helprendszert.

A TopSpeed-termékek szinte hemzsegnek a technikai újításoktól. Minden fordító olyan egyedi előnyöket kínál, amelyeket egyetlen más gyártó sem tud felmutatni. Sokan mondják el róluk: csodálkoznak, hogyan is tudtak eddig meglenni, e szolgáltatásokat nélkülözve. Számos módszer immár a múlté.

A TopSpeed fejlesztői környezetet egyszer kell megvásárolni DOS vagy OS/2 alá, s ezután tetszés szerinti nyelvi fordítót azonnal használatba tudunk venni. Az egyes nyelvjárások jellemző poszt- és preprocesszorai (mint például a JPI szupergyors MS Windows resource compilere) is azonnal integráns részként illeszkednek a TopSpeed Environmentbe. Minden egyes fordítót megvehetünk külön is, önálló modulként, de ugyanígy hozzájuthatunk az egyes nyelvek függvénykészletének teljes forráskódjához is.

Modulárisan, nyitottan, redundancia nélkül

Bármelyik TopSpeed nyelvvel közvetlenül fejleszthetünk DOS, Windows és OS/2 Presentation Manager alá felhasználói programokat. Helyesen beállítva a compiler változtatható opcióit (amelyekkel kikapcsolhatjuk a TopSpeed-specifikus nyelvi bővítéseket) akár nagyszámítógépre is fejleszthetünk hordozható programokat.

Mindmáig egyetlen létező C++-implementáció sem volt képes megoldani a redundáns kód elkerülésének alapproblémáját. Primitív fordítókat használva az eljárástáblázatok többször 10 kbájtnyi, nem kívánt kódot kényszerítettek feleslegesen a futtatható EXE állományba. Ennek a TopSpeedben vége! A SmartClass linkelési eljárás garantálja, hogy kihasználatlan osztályok, statikus és nem virtuális eljárások, amelyekre sohasem kerülne a vezérlés, kimaradnak a végleges programból. A többi C++-compiler arra kényszerít bennünket, hogy vagy újratervezzük könyvtáraink tartalmát, vagy együttélünk azzal a tudattal, hogy programjaink felesleges kódsorozatoktól terheltek.

Overlay — kód és adatok

A TopSpeed szegmensalapú overlay rendszere (BOOST) mind a futtatható kódot, mind pedig az adatot is overlayzi! Egyetlen más overlay rendszer sem biztosítja ezt számunkra DOS alatt. A BOOST használata egyszerű: csak ki kell választanunk a kívánt overlaymodellt, a többi a TopSpeed dolga. A BOOST-hoz tartozó teljes programozói felületet (application programming interface-t) is megkapjuk természetesen. A TopSpeed BOOST áttöri a DOS emelte korlátokat: 16 Mbájtig tetszőleges méretű programokat hozhatunk létre. Ha ezt a lehetőséget kombináljuk a DLL-ekkel (lásd alább), akkor szinte korlátlan a programfejlesztés.

A projektmenedzsment újragondolása

A TopSpeedben bevezetett intelligens linkelési technológia messze több új lehetőséget ad, mint a hagyományos linkerek. Egyszerűen fogalmazva: ha egy rutin vagy adat nem kellhet a program futása során, akkor ez(eke)t egyszerűen bele sem teszi a végleges programba a linker.

Néhány egyéb, egyedi kódgenerálási technikát alkalmazva a TopSpeed nyelvek képesek a szoftveripar lehető legtömörebb és egyúttal leggyorsabb, végrehajtható EXE állományainak létrehozására.

Használjuk a TopSpeed smart linking technológiáját, és a továbbiakban már soha nem lesz szükségünk arra, hogy nagyobb méretű fejlesztéseinket kisebb részekre daraboljuk a könnyebb kezelhetőség reményében. Programunk logikai struktúráját többet nem kell erőszakosan részekre osztani.

A TopSpeed kizárólagos tulajdonságai szabad kezet adnak a programozó számára abban, hogy összetettebb projektjét a lehető legszabadabban, hatékonyabban és elegánsabban szervezze, de egyúttal azt is garantálva, hogy egybájtnyi redundáns kód sem kerül ki a keze alól. A programozó termelékenysége előtt így soha nem remélt új dimenziók nyílhatnak.

A C programozók üdvözölni fogják az ún. TopSpeed SafeLink szolgáltatást, amely fordítási időben ellenőrzi a függvények paraméterátadását és a memóriamodellek helyes használatát. A SafeLink még azelőtt eltávolítja a hibákat, mielőtt azok felszínre kerülhettek volna, így többórás, programteszteléssel töltött fejlesztési időt takaríthatunk meg.

DOS/Multitasking és DDL

Mindegyik TopSpeed nyelvben beépített támogatást kapunk — egy időosztásos ütemező váza (preemtive time-sliced scheduler) révén — arra, hogy egy időben több, azonos erőforrásokat mozgató feladatot hajtathassunk végre programjainkkal DOS alatt.

Minden nyelvhez jár egy speciális multi-thread library, mely véd a DOS nem újrahívható (non-reentrant) funkciói ellen.

A TopSpeed az OS/2-ben és Windowsos platformon bevezetett dinamikusan láncolt függvénykönyvtárak (Dynamic Link Libraries) adta hatékonyságot DOS alá is megvalósította. Anélkül, hogy újra kellene fordítani alkalmazásainkat, kiszolgálhatjuk régebbi ügyfeleinket úgy is, hogy egyszerűen csak a megváltoztatott részét küldjük el lemezen számukra. Ez értelemszerűen futásidőben jut érvényre az eredeti programban.

Az összes TopSpeed nyelv egyazon kódgenerátoron és run-time libraryn

alapul. Ez egyszerűvé teszi a vegyes nyelvű fejlesztést TopSpeed-környezetben. A modernebb nyelvek, mint például a Modula-2 rengeteg kiegészítést vehetnek át korábbi, gazdag C és Pascal fejlesztői kiegészítésekből. Egy nem homogén programozói team tagjai saját kedvenc TopSpeed nyelvjárásukat használhatják, nem kell áttérniük egy közös nyelvre; a TopSpeed többnyelvű fejlesztés esetén is garantálja az optimális kódot.

Modula-2 és Pascal valódi objektumokkal

A TopSpeed Modula-2 és Pascal támogatja az objektumorientált programozáshoz nélkülözhetetlen tulajdonságokat, mint a többszörös öröklés (multiple inheritance), automatikus objektum-inicializálás, adattípus-védelem, típusellenőrzés; ezek a mai OOP-szabványnak elengedhetetlen részei. Ráadásul a TopSpeed Modula-2, Pascal és C++ kompatibilis egymással — mind objektum, mind pedig osztály (class) és bináris szinten —, így bármelyik nyelvben kifejlesztett szoftvermegoldás felhasználható a másik kettőben is.

A TopSpeed C 3.0

C — „megfejelve"

A Jensen & Partners International (JPI) jóvoltából egy száz százalékig az ANSI X3.159-1989 ajánlásoknak megfelelő C-implementáció készült DOS és OS/2 alá.. Ha fejlesztésünk azt kívánja meg, hogy portábilis, azaz hordozható kódot írjunk, akkor ne feledkezzünk meg a TopSpeed C-ről.

Ha már valaki a C nyelvet választotta annak hatékonysága és rugalmassága miatt, itt a kedvező lehetőség, hogy bebiztosítsa magát: végrehajtható programja a lehető leggyorsabb lehet, minimális memóriaigénnyel. A TopSpeed C (a továbbiakban TSC) segítségével gyors és kompakt bináris állományokat kaphatunk C forráskódunkból. A TSC-t a C-fordítók sorában a legjobbak közé rangsorolják.

A régi forrásprogramok is megíjfodnak: a Microsoft C vagy Borland Turbo C források közvetlenül lefordulnak a TSC alatt is — esetleg minimális odafigyelés kell néhány fordítási opció megfelelő beállításához. Áthozva C kódunkat a TopSpeed többnyelvű integrált fejlesztői környezetébe, belekóstolhatunk mások TopSpeed Pascalban, Modula-2-ben vagy C++-ban írt fejlesztéseibe is.

Intelligens linkelés: csúcssebesség

A TSC simán elintézi az óriási méretű programok szerkesztését a Smart Linking technológia segítségével. Kizárólag azok a könyvtári függvények kerülnek csak bele a futtatható programba, amelyekre hivatkozás van (bárhol a programfolyamban).

Használjuk a TSC linking-technológiáját, és a továbbiakban már soha nem lesz szükségünk arra, hogy nagyobb méretű fejlesztéseinket kisebb részekre daraboljuk a könnyebb kezelhetőség reményében. Programunk logikai struktúráját többé nem kell erőszakosan részekre osztani.

Based pointerek, kód- és adatoverlay

A JPI vezette be először a szegmensrelatív pointerek fogalmát. Ezek a a szegmensen belüli mutatók relokálható, relatív adatszerkezetet tesznek lehetővé (ilyen például a heap is).

A típusellenőrzést is végző biztonságos link funkció, futásidejű nil pointer konzisztenciavizsgálat, tömbök indexelésének, a stack túlcsordulásának folyamatos figyelemmel kísérése —

Tulajdonságok, képességek, árak

— Paraméterátadás a processzor regiszterein keresztül a nagyobb sebesség és a minimális veremfelhasználás érdekében.
— Feltételes fordítási lehetőség, mely lehetővé teszi különböző változatok és tesztverziók kezelését.
— Inline gépi kódú sorok (assembler kód a normál forrássorok között).
— Abszolútváltozók kezelése, amelyeket a program a betöltésekor előre megadott helyre tesz a memóriában.
— Egyszerre 32 konkurens program futhat; dead-lock-elkerülő algoritmus.
— Szegmensen belüli, ún. short pointerek.
— Standard TopSpeed compilervezérlő (pragma) utasítások.
— Debuggolás vagy a TopSpeed VID-jével (Visual Interactive Debugger), vagy kérésre Microsoft CodeView-kompatibilis módon.
— 80x87-es koprocesszor támogatása.
— Hívási konvenciók ellenőrzése a mixed-language fejlesztéshez.
— Microsoft .OBJ kódkompatibilitás.
— Automatikus prototípus-generálás.
— Preprocesszor-output.
— Volatile adattípus. (Olyan adatok, amelyekre ha egyszer volt már hivatkozás, akkor utána rögtön eltűnnek, vagyis felszabadul a memóriaterület, amelyet addig lefoglaltak.)
— Definiálható figyelmeztetési szintek a compiler számára.
— Mixed memóriamodell is használható.
— Állandó stack- és heapméret-figyelés.
— Tárrezidens programok írása (interrupt handlerek).
— 80 bit hosszú, dupla pontosságú valós számok kezelése.
— Kompatibilitás a Turbo C 2.x és Microsoft 5.1 és 6.0, valamint Quick C programokkal.
— DOS és OS/2-es változatban is kapható.
— Az OS/2 változat real és protected módú programot is létre tud hozni.
— Windows 3.0 kód generálható, ha megvan a TopSpeed TechKit.
— Kompatibilitás a Borland BGI-vel.
— Library Source Kit (ehhez is kell a TopSpeed TechKit).
— ANSI Standard Library.
— Unix-kompatibilis funkciók.
— JPI karakterbázisú ablakkezelő függvénykönyvtár.
— Borland Clipping Window Library.
— JPI Process Library.
— Microsoft-kompatibilis grafikus rutinok könyvtára.
— Teljes MS-DOS interfész.
— A teljes Microsoft és Borland speciális függvénykészlet (ha az azonos funkciójú függvények eltérő nevűek, mindkettőt tartalmazza a TSC).
— Microsoft egérkezelő rutinok.

Árak az Alaplap Postán keresztül:
TopSpeed C Compiler 12 000 Ft + ÁFA
TopSpeed C Standard Pack (Compiler + TS Environment) 24 000 Ft + ÁFA
TopSpeed C Professional Pack (Standard + TechKit) 36 000 Ft + ÁFA
TopSpeed C Library Source Kit 12 000 Ft + ÁFA

IBM PC

SOLARSOFT KATALÓGUS

Lemezszám: 527

Név: PCX-SHOW 4.0

Szerző: PC WEST, 1991.

Leírás: Grafikus program, PCX formátumot kezel.

A program PCX formátumú állományokkal dolgozik. Demóját elindítva szinte minden lehetőségéről képet kapunk. Nemcsak szemgyönyörködtető, hanem igen szellemes is a képek egymásutánja, az angol kísérőszöveg. A demo „forrásszövege" olyan egyértelmű, hogy különösebb előképzettség nélkül is bárki megírhatja saját ízlése szerint a hasonló programot. Ne ijedjünk meg, ha indításkor azonnal a demóval kezdl Ennek az az oka, hogy ha nem adunk meg paraméterként állománynevet, alapértelmezés a demoállomány.

A shareware verzió az alábbi fontokat ismeri: HELV12, HELV8, OLDENG, TROM1, FUTURE, COMP10, EURO

Konfiguráció: 348 K szabad memória, EGA, VGA vagy SVGA grafikus kártya 256 K képmemóriával. Nyomtató és színes monitor ajánlott, de nem kötelező.
Alapértelmezésben 640 x 350 felbontású EGA képpel dolgozik.

Lemezszám: 528

Név: WitchBall

Szerző: Ralf Mohmeyer, Proline Software, Németország, 1991.

Leírás: Memory-szerű játék.

A játék lényege: az egyforma képek magtalálása, ezt a pályák különböző elrendezése (újraépülő falak, befalazott képek stb.) nehezítik.

Konfiguráció: 512 kB RAM; VGA grafikus kártya.

Lemezszám: 529

Név: Bartender 1.51

Szerző: Kevin Vigor, USA, 1990.

Leírás: Vonalkódnyomtató segédprogram.

Alkalmazói programokból is indíthatjuk. Memóriarezidens, 10 különböző vonalkódtípust kezel, 17-30 K memóriaigényű. Kisméretű, gyors, a program jelenléte nem lassítja a nyomtatást. Menüvezérelt, bárhol kérhetünk magyarázó szöveget.

Rugalmas: 1/10 inch és 1 inch magasság között 10 különböző vonalkódot tud nyomtatni. Egy lapon keresztbe hat vonalkód lehet. Szövegszerkesztőnkkel, adatbázisunkkal, számolótáblánkkal, más olyan rendszerrel, amely outputot nyomtat, használható anélkül, hogy megszokott munkafolyamatunkon változtatnunk kellene.

Az alábbi nyomtatótípusokat támogatja: PostScript, Epson, IBM ProPrinter, HP LaserJet 150 dpi-vel, HP LaserJet 300 dpi-vel, Epson 24 tűs, Toshiba 24 tűs, Okidata MicroLine.

Az említett tíz vonalkódtípus mindegyike más célt szolgál: UPC-A, UPC-E, EAN/JAN-12, EAN/JAN-8, 3 of 9, Extended 3 of 9, Interleaved 2 of 5, Code 128, Codabar, Zip + 4.

Lemezszám: 530

Név: Dungeons

Szerző: Enque Software, illetve Markus Mueck, 1991.

Leírás: Két játék.

Két játékos játszhatja a tankcsatát. Egyikük az IRAKIAKAT, másikuk a SZÖVETSÉGESEKET képviseli.

A harcmező digitalizált, a tankok sebessége változtatható, vannak kommunikációs tornyok stb. Telefonvonalon is játszhatunk ellenfelünkkell (Ez azért Magyarországon meggondolandó.)

Indításkor beállíthatjuk a játék sebességét (1—9). A bejelentkező képernyő után számos, a játék menetére vonatkozó információ jelenik meg.

Célunk az ellenfél megsemmisítése. Számos tankkal rendelkezünk, tartalék üzemanyagunk és töltényünk is van a saját térfelünkön. Célozzunk jól, és már lőjünk is!

Érdemes először az ellenség bunkereit, majd pedig a lőszerraktárait és üzemanyag-tartalékait megszereznünk. Ezek érik ugyanis a legtöbb pontot.

Ne menjünk az eltalált üzemanyag-(fegyver-)tároló helyek közelébe tankjainkkal, ezek ugyanis bármikor tüzet foghatnak.

A Magic Miner klasszikus ugró-futó játék. Célja, hogy mind a 7 szintet teljesítsük 3 életből. Ez a 3 élet azt jelenti, hogy háromszor érinthetünk meg szörnyet, lézert vagy más, halált hozó dolgot. Harmadszorra angyalok kórusa kíséri a megboldogultat.

Ha például valahol a negyedik szinten halunk meg, akkor ezt a szintet kell elölről kezdenünk.

Hogyan teljesíthetjük az egyes szinteket? Minden szinten három dolgot kell begyűjtenünk, majd pedig meg kell szereznünk a kulcsot, amely a következő szinthez vezet.

Konfigurációs igény: 12 MHz-es AT ajánlott, EGA/VGA grafikus kártya/monitor, 212 K memória.

Lemezszám: 531

Név: Peresztrojka

Szerző: Leonid Schneider.

Leírás: Játék.

Az ötlet az egykori szovjetunióbeli állapotokon alapul: túl kell élni a zűrös körülményeket, anyagi javakat kell szerezni, amennyit csak bírunk, el kell kerülnünk a párt bürokratáit...

A kezdő képernyő 15 másodperces orosz dallal indul, a zenét nem tudjuk megszakítani. (De miért is akarnánk?) Ennek befejezésekor jelenik meg a játékszabályokat ismertető képernyő, stílusosan oroszul.

Szürke ellipszisszerűség — Törvények és szabályzók, amelyek változnak, eltűnnek, újak keletkeznek teljesen váratlan helyeken. Ezeket kell arra használnunk, hogy följussunk a képernyő jobb felső sarkába.

Zöld béka — a játékos, aki demokrata.

Vörös béka — a bürokrata párttagok (őket el kell kerülnünk).

Kék labda — fogyasztási cikkek (elég ritkák).

Vörös labda — kemény valuta (még ritkább).

Sárga labda — progresszívan növekvő adó (ezt is jó elkerülni).

Sárga körben szám — a „Nagy Újjáépítési Út" mérföldköve. 25 mérföldkövet kell összegyűjtenünk. Az egymást követő pályák egyre nehezebbek.

Enter leütésekor indul a játék, Esc felfüggeszti.

Három esélyt kapunk arra, hogy túléljük a Szovjetuniót. Ha vesztesek vagyunk, akkor a párttag bürokraták (a vörös békák) eljárják saját kis diadaltáncukat, miközben részvétüket fejezik ki. Amíg a tánc tart, '0'-val kiléphetünk a játékból, '1'-gyel pedig ismét szerencsét próbálhatunk.

Konfiguráció: EGA- és VGA-kártya és 356 K szabad memória szükséges a játékhoz.

Lemezszám: 532

Név: Oilcap 4.0

Szerző: William Soleau, 1991.

Leírás: EGA-játék.

A játék lényege: versenyfutás az idővel, hogy a kiömlő olajat fel tudjuk fogni egy általunk építendő csőrendszerbe. A képernyő egy 11 x 8 elemű négyzetrács. Szintenként induláskor megjelenik valahol egy START felirat, innen fog az olaj ömleni, amikor az időzítés lejár. A képernyő alsó részén (a játéktér alatt) négy kis négyzetben csődarabok láthatók. Egerünkkel úgy kell elhelyeznünk ezeket a felső táblára, hogy az olaj semmiképpen ne folyjon ki a csővezetékből. Az első csődarab elhelyezése után alul ennek helyére lép a következő, a bal szélső négyzetbe újabb csődarab lép be. Folyamatosan tehetjük fel ezeket a táblára, akkor is, amikor az olaj már elkezdett folyni. Lényeg, hogy gyorsabban épüljön a csőrendszer, ne folyjon ki az olaj. Vannak olyan csőrészletek (mindig megjelölve), amelyekért bónuszpontok is járnak. Nem helyezhetünk csöveket sem a fekete kockákba, sem pedig a START-ra. Helyezhetünk azonban egy már lerakott kapcsolóelem tetejébe, ilyenkor azonban 4-5 másodpercig nem rakhatunk ki újabb csövet. Csak akkor kerülünk ki győztesen a játékból, ha az olaj már végképp nem tud semerre sem folyni.
Az olaj sebességét állíthatjuk. Kezdetben 40 másodpercünk van arra, hogy feltegyük csőrendszerünk első elemeit, mielőtt az olaj megindulna. Ezután már versenyt futunk az olajjal. A magasabb szinteken már ez a 40 másodperc csökken 35-re, 30-ra, 25-re. A 7. szinttől pedig megnő az olaj sebessége is.

Konfiguráció: EGA-grafika, egér.

Lemezszám: 533

Név: CD-Man

Szerző: Creative Dimensions, USA.

Leírás: VGA-játék.

A kis sárga emberke a kurzorbillentyűkkel irányítható. Meg kell ennie kis zöld pontokat, nagy piros pontokat, kulcsot, menekülnie kell rozmárfogú mérgespókok elől.

A játék grafikája egyszerűen gyönyörű. Középkori kertben van ez a labirintusszerűség. Oldalt vízesés, középen szökőkút. Mohos kőfalak, alagút, várkastély, kapurostély... Csilingel a zene, élmény játszani.

Konfiguráció: VGA-kártya, monitor.

Lemezszám: 534

Név: DIET 1.02b

Szerző: Teddy Matsumoto, Japán, 1991.

Leírás: Tömörítő.

Filozófiájában alapvetően különbözik helytakarékosságot célzó (és elérő) társaitól. Itt ugyanis kicsomagolás nélkül használhatók a tömörített állományok: olvashatók az adatállományok, végrehajthatók a .COM és .EXE kiterjesztésűek. A DIET.EXE tehát egyfelől tömörítő segédprogram, másfelől olyan eszköz, amely képes ezekkel a tömörített állományokkal dolgozni. Az igazi újdonság az LZEXE és PKLITE programokhoz képest az, hogy a DIET adatállományok tömörítésére is alkalmas.

Ez a legújabb, 1.02b verzió támogatja a Lotus-Microsoft EMM.sys-t, így rezidens részének csak egy kis töredéke található a 640 K DOS-memóriában. A szoftver kifejlesztésekor a laptop gépek világát tartották szem előtt az alkotók.

Lemezszám: 535

Név: ARCTOOL 7.0

Szerző: G. D. Davis, GDSOFT, USA, 1991.

Leírás: Tömörítő keretprogram.

Menüvezérelt keret, amely az összes népszerű tömörítőprogramot ismeri. Kezeli a PKZIP, ARC, PKARC, LHARC, LHA, LHS, ARJ, ZOO, SQUEEZ és PAK tömörítőkkel létrehozott állományokat.

Ez a legújabb, Arctool 7.0 verzió már a PAK, ARC, LHARC és PKZIP programok SFX formátumú outputjaival is dolgozik. Ezek az önkicsomagoló állományok .EXE vagy .COM kiterjesztésűek. Az Arctool megvizsgálja az összes ilyen kiterjesztésű állományt, és ha SFX-nek bizonyulnak, ugyanúgy kezeli őket, mint a többi tömörített állományt.

Mivel az Arctool csak a keret, lemezünknek tartalmaznia kell azokat a kicsomagoló szoftvereket, amelyekkel az állományokat kibontjuk.

Az állományokat tetszőleges sorrendbe rendezhetjük. A megjelölt állományt a FIND parancs hatására megkeresi a lemezen.

A NEW utasítással egy másik archív állománnyal foglalkozhatunk.

További szolgáltatásai:

— Új grafikus interfész.

— Az archív állományokban található archívok kicsomagolása, 4 szintes mélységig.

— Rendszeren belül másolhatók, átnevezhetők és törölhetők az archív állományok.

— Vírusellenőrzésre saját megszokott víruskeresőink használhatók.

— Könyvtárak módosítása és létrehozása a rendszerből való kilépés nélkül.

— Az archív hozzáférési utakhoz konfigurációs állomány készülhet.

— A tömörített állományok számát csak a rendelkezésre álló memória korlátozza. (Korábban a korlát 250 volt.)

— NetBIOS-, DeskView-, Windows-kompatibilitás.

Lemezszám: 536

Név: Magic&Win

Szerző: Fran Vong, Németország

Leírás: Négy EGA-játék.

MAGIC-BALL: lényege, hogy az egész pályát kitöltsük a biliárdgolyókkal. Ezt nehezíti, hogy ha három egyforma színű kerül egymás mellé, akkor eltűnnek, illetve az, hogy az egyes átjárókat könnyen elzárhatjuk.

MAGIC-COBRA: a kígyónak meg kell ennie a pályán található összes „egeret". Ezt akadályozza az egerek elhelyezkedése, a pálya felépítése, és az, hogy a kígyó a falánkságától egyre nő.

MAGIC-OLYMP: a fellelhető összes nektarint (őszibarackszerűség) meg kell ennünk. Akadályoznak az ellenséges manók, és a magasból lehulló barackok.

WIN: célunk, hogy az összes dioxin-szörnyet megsemmisítsük (a kockákat rájuk toljuk).

A játékok jó grafikával készültek. Részletes, német nyelvű leírás található a MAGIC.TXT fájlban.

Lemezszám: 537

Név: Dame40

Szerző: Jürgen Schmitt, Németország, 1991.

Leírás: Játék — bármely grafikus kártyához.

Klasszikus dámajáték, amelynek lényege, hogy az ellenfél korongjait átlós irányban átlépve, minél többet „kiüssünk". A 4.0-ás verzió különlegességei: a két- és háromdimenziós kivitel; választható színek, játékmódok, stratégiák; kinyomtathatóság; SoundBlaster kártya és egér csatlakoztatható.

Windows alá installáláshoz a WINDOW.TXT nyújt segítséget.

Lemezszám: 538

Név: Quadrax

Szerző: Proline Software, Németország.

Leírás: Játék bármely grafikus kártyához.

A játék lényege: az egyforma, számozott kockák eltüntetése. A játék elején kis bemutató tájékoztat a lehetőségekről. Német nyelvű.

Lemezszám: 539

Név: ASIC30

Szerző: David A. Visti, USA, 1991.

Leírás: BASIC nyelvjárás.

Az ASIC egy BASIC programozási nyelv IBM PC-re. Integrált editort tartalmaz, ebből editálhatjuk, fordíthatjuk, tesztelhetjük programjainkat.

Az ASIC a BASICA és a GWBASIC nyelvek alapján készült, 80-nál több BASIC utasítást támogat, integer- és stringváltozókat, stringsorozatokat kezel. Fordítása gyors, az általa generált kód tömör. Az ASIC utasítások csaknem mindegyike megtalálható a BASICA és a GWBASIC nyelvekben is.

Használatához 305-405 K szabad memória szükséges. Maga az ASIC fordító 305 K

ALAPLAP
CSOMAGKÜLDŐ SZOLGÁLAT
POSTA

Jelenleg mintegy 1000 szoftverből, illetve külföldi szakkönyvből válogathat. Az aktualizált lista megtalálható mostani számunk lemezmellékletén. A megrendelt szoftvert vagy külföldi szakkönyvet postai utánvétellel 2 héten belül házhoz szállítjuk.

MEGRENDELÉS

Megrendelem postai utánvétellel az alábbi termékeket. A vételárat és a postaköltséget átvételekor kifizetem.

A) SZOFTVEREK:

....

....

....

B) SZAKKÖNYVEK:

....

....

....

Dátum:

(aláírás)

MEGRENDELŐLAP

Megrendelem utánvétellel az Alaplap kiadványsorozataiban megjelent alábbi műveket:

ALAPLAP KÖNYVEK

... pld: Nagy Gábor: Tömör gyönyör	256,–
... pld: Kis János–Szegedi Imre: Új víruslélektan	256,–
... pld: Kis János–Szegedi Imre: Vírushatározó	256,–
... pld: Jodál Endre: Általános fogalmak (Számítástechnikai alaplexikon I., 2. kiadás)	356,–
... pld: Jodál Endre: Adatkommunikáció és számítógép-hálózatok (Számítástechnikai alaplexikon II.)	356,–
... pld: Farkas Ernő: PC-szótár	456,–
... pld: Kis János: BBS — avagy az elektronikus postaláda (lemezmelléklettel)	656,–

ALAPLAP FÜZETEK

... pld: Detrik Péter: Az SQL nyelvről	375,–

ALAPLAP LEMEZEK

... pld: Bliss főkönyvi program (demó és leírás)	750,–
... pld: Norton Guide keretprogram (leírás)	500,–
... pld: PathMinder segédprogram (leírás)	500,–
... pld: CSProlog nyelv (leírás)	1000,–
... pld: LIM EMS 4.0 memóriakezelő (leírás)	1000,–
... pld: Magyar betűkészletek Windows 3.0-hoz	1000,–

PC Turbo Klub

VÁLASZLAP

Eddigi PC Turbo Klub-tagoknak nem kell kitölteniük és beküldeniük, ha címüket az új címadatbázisban is szerepeltetni akarják. (Lásd erről részletesebben cikkünket az Alaplap 92/8. számának 54. oldalán.)

☐ A PC Turbo Klub tagja vagyok, de kérem, hogy az új címadatbázisban címem már ne szerepeljen.

☐ Nem vagyok a PC Turbo Klub tagja, de szeretném, ha túloldalt megadott címemet az új címadatbázisba felvennék.

Tudomásul veszem, hogy a PC Turbo Klub új címadatbázisába való felvétel esetén címemet az Alaplaptól szakmai címanyagot kérő cégek és intézmények direct mail akcióihoz (meghívók, prospektusok stb. kiküldéséhez) felhasználhatják.

Dátum:

....
aláírás

INFORMÁCIÓKÉRÉS

Kérem, hogy az itt általam **BEKARIKÁZOTT KÓDSZÁMÚ** hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61	62	63	64	65
66	67	68	69	70
71	72	73	74	75
76	77	78	79	80

Beküldhető: 1992. november 30-ig

ALAPLAP 1992/10 OKTÓBER

FELADÓ:

A) Egyéni érdeklődő:

Név: ....

Utca, házszám:..

Helység: ....

Irányítószám: ....

B) Vállalati érdeklődő:

Cégnév:..

Ügyintéző:...

Utca, házszám:..

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Belföldön díjmentesen feladható

Cédrus Kiadó
Pf. 74

Budapest

1441

FELADÓ:

Név: ....

Cég: ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Belföldön díjmentesen feladható

Cédrus Kiadó
Pf. 74

Budapest

1441

FELADÓ:

Név:...

....

Foglalkozás/Beosztás: ....

Cím: ....

....

Helység:...

Irányítószám: ....

Belföldön díjmentesen feladható

Cédrus Kiadó
Pf. 74

Budapest

1441

FELADÓ:

Név: ....

Cég: ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Belföldön díjmentesen feladható

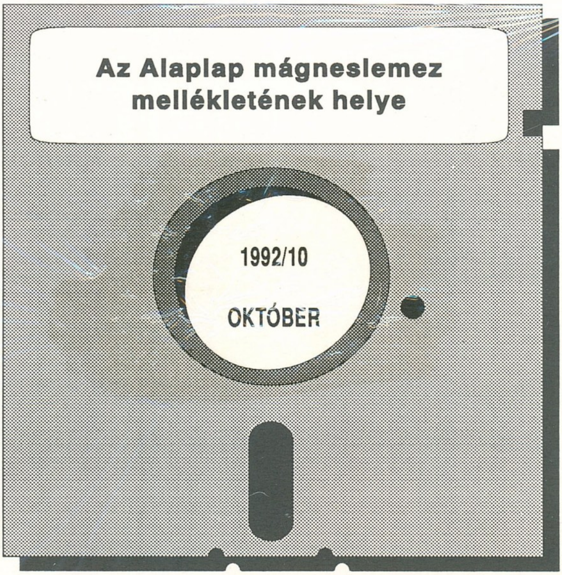
Cédrus Kiadó
Pf. 74

Budapest

1441

A LEMEZMELLÉKLET TARTALMA:

- Példaprogramok a tárgyorientált programozáshoz
- Memóriatérképek összehasonlítása
- Kimenetek — egymással felcserélve
- McAfee antivírus programjainak felfrissítése
- Lemezinformációs és ellenőrző program
- OrCad II. nyalánkságok
- Memóriajáték a színekkel
- A számrendszeres fejtörő megfejtése
- Snobol a betűgyakorisághoz — egy kis költészettel



memóriát igényel, az editornak 200 K-nál kevesebb az igénye. 405K akkor szükséges, ha az integrált editorból akarunk fordítani és hibát keresni.

Lemezszám: 540

Név: Fallout

Szerző: Richard Schabath, 1990, illetve Gisbert Ehlert, 1991.

Leírás: Két EGA-játék.
C64-re és Amigára készült elődöktől (BOULDERDASH és WERNER) származik a MEGABOULDER 2.0 játék ötlete. Célunk, hogy szintenként a lehető legtöbb gyémántot összegyűjtsük. 1000 pont elérésekor kerülhetünk felsőbb szintű pályára. Nemcsak gyémántgyűjtéssel juthatunk pontokhoz, hanem kis szörny megölésével, sörösüvegek begyűjtésével is. A teljesített szintek után is jár időbónusz.
Állítható a játék sebessége, a hang ki/be kapcsolható. Saját editorával módosíthatjuk a 21-99 szinteket. DR-DOS 3.41 és MS-DOS 3.3 vagy magasabb verzióval játszhatunk.
A FALLOUT 1.0 szerzője két játék ötletét variálta a FALLOUT kifejlesztésekor.
Az egyik a TETRIS. Kis kockákból álló láncok (3 eleműek a kezdőknek, 4 eleműek a haladóknak) jelennek meg. A láncot alkotó kockák színe különböző, a láncok lehetnek vízszintesek, függőlegesek, ferdék.
A leérkező láncok azonos színű kockái kioltják egymást, eltűnnek. Menet közben a lefelé tartó lánc színeit forgathatjuk, magát a láncot eltolhatjuk.

Lemezszám: 541

Név: EGA Games #3

Szerző: Fran Vong, illetve Richard Schabath.

Leírás: 2 EGA-játék.
A MAGIC-BALL lényege, hogy az egész pályát kitöltsük a biliárdgolyókkal. Ezek a golyók láncokban jönnek. A láncok hossza változó — 6 és 28 között változhat véletlenszerűen. Nehezíti a dolgunkat, hogy ha három egyforma színű golyó kerül egymás mellé, akkor eltűnnek, illetve az is, hogy a pálya egyes átjáróit könnyen elzárhatjuk. A láncokat kurzorbillentyűkkel irányíthatjuk.
Játékonként 30 láncunk van. Akkor veszítünk, amikor már olyan lánc jönne, amelyik nem tud bejutni a pályára.
Az 540-es lemezen lévő MEGABOULDER 2.0 erre a lemezre is rákerült.
Konfiguráció: EGA/VGA kártya.

Lemezszám: 542

Név: HYPER, DWC, COMPACK

Leírás: Három tömörítő.
Igazi nemzetközi szerzőgárda tömörítőit adjuk közre ezen a lemezen.
A HYPER 2.50 német szerzőpárosa Peter Switzki és Klaus Peter Nischke. Az 1990-es évben készült verzió jó hatásfokú és nagyon-nagyon egyszerűen kezelhető. Mindössze 10 utasítást kell megtanulnunk, ugyanakkor a tömörítés minimumszabványnak teljesen megfelel. Mindazok, akik csak használni akarnak egy tömörítőt — nincsenek speciális, különleges elvárásaik —, pillanatok alatt elsajátíthatják kezelését, nem is kell más tömörítővel foglalkozniuk.
A DWC 5.01 szerzője Dean W. Cooper, USA, 1988. Az LZW (Lempel-Ziv-Welch) algoritmust használta alapul. Ez a tömörítő a szöveges állományokat több mint 75%-kal nyomja össze. Kezeli a tárgykódot, könyvtárakat, végrehajtható állományokat, grafikus bitmapeket.
A COMPACK 4.4 kifejlesztője a Prominence Computer Services Ltd. olasz cég (1991). Elsősorban az .EXE és .COM kiterjesztésű végrehajtható állományok méretcsökkentését célozták meg.

Lemezszám: 543

Név: As-Easy-As 5.0 & LHA

Szerző: TRIUS Inc., USA, 1991

Leírás: Táblázatkezelő
— Nagyméretű táblázatok (8.192 sor — 256 oszlop)
— Menüvezérelt
— Matematikai, statisztikai, logikai, pénzügyi, string-, dátum-, időfüggvények
— Mátrixműveletek
— Állományok összekapcsolása
— Grafikai lehetőségek széles skálája
— Billentyűzetről vagy egérrel vezérelető
— Az egybeszerkesztett szövegek és grafikus részek nyomtatás előtt megjeleníthetők
— 9/24 tűs, HP Laserjet- és HP Pen Plotter-támogatás
— Ablakok! — Hat átméretezhető és elmozdítható munkatábla
— Adatbázis-műveletek, adatbeviteli formátumok, Dbase-állományok írása/olvasása
— Szövegben keresés, helyettesítés
— Több síkban dolgozik (3D szimuláció)
— Makróprogramozási nyelv (70-nél több hatékony makró)
— Makrók lépésenként is végrehajthatók
— Fájlmenedzser
— Támogatja a Hercules/CGA/EGA/AT&T/VGA grafikus kártyákat
— DOS-keret
— Windows 3.0 környezetben is működik DOS-alkalmazásként
— Automatikusan felismeri az EMS-t, virtuális memóriával dolgozik ennek hiányában.
Konfiguráció: IBM PC tetszőleges grafikus kártyával, 384 K szabad memória.

Lemezszám: 544

Név: Windows Games/1

Leírás: 11 játék Windows alá: 5GEWIN, ATTAXX, BACKG002, BANGBANG, BATTLE10, BG060, BLACK024, BLITZ12, BRICK, CHECK11, CHESSNET.

Lemezszám: 545

Név: Windows Games/2

Leírás: 12 játék Windows alá: CHESS31, CHOMP11, CUBIC, GOLF11, HEXTRIS, MEGABRAI, PUZZLE12, SENSOWIN, SHUFFLE, SLOTMACH, SLOTZI, SPACEWAL.

Lemezszám: 546

Név: Windows Games/3

Leírás: 6 játék Windows alá: SCW103, SMFPOKER, SNAKES, SYMBOL-R, TETRIS, TRIPLETS.

Lemezszám: 547

Név: Windows Games/4

Leírás: 3 játék Windows alá: SPGAMES, WINTRIS, YACHT (+ VBRUN100).

Lemezszám: 548

Név: Windows Games/5

Leírás: 3 játék Windows alá: SHITAO11, WINLOTTO, WINPUZZ.

Lemezszám: 549

Név: Windows-segédprogramok/1

Leírás: 8 program + 3 játék Windows alá.
ADLIB-PM
AT11 — Application Timer
BANGBANG — Játék
BARTEYES — Egérkövetés, képernyőmentés
BASICICO — 13 ikon Basic programokhoz
BG060 — Backgammon játék
BJ11 — BlackJack játék
BLANKER — Képernyőmentés
CASELN30 — CaseLiner — kazettafeliratozás
CLIPICO — Ikongyűjtemény Clipperhez
COOKIE — Viccprogram

Lemezszám: 550

Név: Windows-segédprogramok/2

Leírás: 7 program Windows alá.
DBROWSER — Adatbázis-betekintő
DNGR20 — Word for Windows-makrók
ELANA11 — Rendszerkonfiguráló
ENUMW
FOLDER32 — Programmenedzser
NOTEBK12 — Jegyzetfüzet
NOTEMAN — Jegyzetfüzet

Lemezszám: 551

Név: Windows-segédprogramok/3

Leírás: 3 program Windows alá.
IEDIT32 — Ikoneditor
LI — Fájlbetekintő
NAVIG051 — Navigátor

Lemezszám: 552

Név: Windows-segédprogramok/4

Leírás: 9 program Windows alá.
PCPSAVER — Képernyőmentés (német nyelvű)
QUOTE11 — Idézetek minden napra
RAISE111 — Programkapcsoló segédprogram
SHOWGIF — Image-betekintő/Katalógus
SNAGIT16 — Képernyőfotózó
STARTUP — A bejentkező logo megváltoztatása
TELEPIF — Adatátvitel

WGIF10B — GIF betekintő
WIDEBUG — Hibakereső fejlesztőknek

Lemezszám: 553

Név: Windows-segédprogramok/5

Leírás: 3 program + 1 játék Windows alá.
TASKM104 — Task Manager
WICK20D — Csekk-könyv
WINP — Póker
HOTPOP10 — Menürendszer

Lemezszám: 554

Név: Windows-segédprogramok/6

Leírás: 6 program + 1 játék Windows alá.
COPYIT13 — Fájlmenedzser
WBT30A — Fájlmenedzser
WINLOAD — Windows-indító
WINP — Póker
WINPOST — Memókezelés
WINRES10 — VGA grafikus meghajtó (DOS)
WINWHERE — Állománybetekintő (német)

Lemezszám: 555

Név: Windows-segédprogramok/7

Leírás: 7 program Windows alá.
WINNEC24 — NEC 24 tűs nyomtatóhoz meghajtó
WINREAD — Offline postaolvasó
WINRUN21 — Programindító
WINZIP10 — ZIP-menedzser
WPEDIT — Editor programozóknak
WINBMPS — 4 jó BMP-képző
ZORTECH — Zortech fejlesztőkönyvtár

Lemezszám: 556

Név: Windows-segédprogramok/8

Leírás: 4 program + 2 játék Windows alá.
ACHART11 — ASCII/EBCDIC kódtábla
ADDRESSEN — Win-címek (német)
ALFRED11 — PopUp menü
ANTS — Viccek képernyővel
ATTAXX — Stratégiai játék
BBVIEW12 — Grafikusfájl-betekintő

Lemezszám: 557

Név: Windows-segédprogramok/9

Leírás: 4 program + 2 játék Windows alá.
ARACHNID — Kártyajáték
BARTEYII — Egérkövető/képernyőmentő digitalizált hanggal
BCKMEN20 — Menük
BIGDSK22 — 2048 X 2048 desktop multitaszking
BITMAN10 — Bitmap-betekintő/módosító (EXE/DLL)
BLITZER — Helikopterjáték

Lemezszám: 558

Név: Windows-segédprogramok/10

Leírás: 6 program + 3 játék Windows alá.
BLKOUT22 — Játék
CHOMP11 — PacMan játék
CONMAN — Hálózati illesztő Windowshoz
CUBIC — 3D kockajáték
DAGENT14 — Double Agent-fejlesztő
DC110 — DiskCopy másolóprogram
DEUS099 — Deus ex machina (Mac-klón)
DH060 — Desktop Helper
DHELPER — Desktop Helper

Lemezszám: 559

Név: Windows-segédprogramok/11

Leírás: 10 program Windows alá.
DSIZE10 — Utanként mutatja a bájtokat (szuper!)
DSKTOP13 — Programmenedzser
DUPICON — (DOS) megtalálja a dupla ikont
EARTH — Forgó földgömb
ESYMEN30 — Easy Menu
FINANCE — Pénzügyi program
FONTER40 — Kinyomtatja az összes rendelkezésre álló fontkészletet
FUSE — Animált BMP kép
GALLOMAK — Gallo makrók (Word for Windows + Laser)
GRAVITY — Gravitációszimulátor

Lemezszám: 560

Név: Windows-segédprogramok/12

Leírás: 13 program Windows alá.
HSCRBLINK — Screen Blanker
ICNMAGIC — Ikoneditor
ICNMAN11 — Ikonmenedzser
ICNMASTR — Ikonmenedzser
ICOART — Ikonkönyvtár 2X28-as ikonokkal
ICONDLL — IkonDDL ikonkönyvtárat hoz létre (dos)
ICONEXT — EXE/DLL-ből megnéz/kivesz ikonokat
ICONFIX — Közlekedőprogram DOS ikonokhoz
ICONMAN — Ikonmenedzser (ikonok kivitele/ beillesztése)
ICONPEEK — BMP-t hoz létre minden ikonhoz (DOS)
IINJECT — Ikoninjektor (DOS)
IKE — Ikoneditor 32 X 32 BITMAP-HEZ
ISLAND21 — 3D térkép fraktálokkal

Lemezszám: 561

Név: Windows-segédprogramok/13

Leírás: 7 program Windows alá.
INIEDIT — INI állományokba betekintés/szerkesztés/nyomtatás
K1 — Naptárrendszer (német)
KALDEMO — Kaleidoszkóp
KERMIT07 — Kermit Windows-hoz
KEYMOUSE — Egér-segédprogram
LASERLBL — Címkenyomtató
LAUNCHUT

Lemezszám: 562

Név: Windows-segédprogramok/14

Leírás: 6 program + 1 játék Windows alá.
PLIFE10 — Játék
PRINTCLP — Clipboard-nyomtató
PROJCLK — Project timer
PSHOP201 — PaintShop
PULSAR — Postscript/PCL átkapcsoló
REFONT11 — Postscript Mac/PC, AFM/PFM (DOS)
REMINDER — Határidő-emlékeztető

Lemezszám: 563

Név: Windows-segédprogramok/15

Leírás: 12 program Windows alá.
MATCH103 — Szöveges Windows állományok összehasonlítása
MATHGRAF — Függvénynyomtatás
MAXMIN — Memóriafelosztás
MOONTOOL — Hold-időszakok
MOREICON — Ikonkönyvtár
MVWIN10 — Állományok ármozgatása, másolása, törlése
NEKO10 — Egérkurzor
NINFO092 — Hálózati információ Excel-támogatással
NJWFIND — Állományokat keres meg, archívokat is
NODOS21 — Ikonkezelő segédprogram, kiszűri a DOS ikonokat
PHMESS — Telefonüzenetek, megbeszélések jegyzetei
PIFHELP — PIF — Help

Lemezszám: 564

Név: Windows-segédprogramok/16

Leírás: 11 program + 4 játék Windows alá.
ROLLEM — Kockajáték
SABDU13 — Másolóprogram
SCRPLAY — Képernyőmentő
SHUFFLE — Játék
SIZE-IT — Képernyőméretet ad meg
SLOTZI — SlotMachine — játék
SMFPOKER — Casino Poker — játék
SNAPAPP — Képernyőfotózó
SNAPSHOT — Képernyő-pillanatfelvétel
SNPSHT11 — Képernyő-pillanatfelvétel
STATLI15 — Státusz-sor
STOP — Igen gyors kilépés a Windowsból
STOPWTCH — Stopperóra
TICKTOCK — Óra
TIM10 — Az idő pénz

Lemezszám: 565

Név: Windows-segédprogramok/17

Leírás: 7 program + 1 játék Windows alá
TOXTOOLS — Toxic tools
TRIPLETS — Játék
UARTMON — RS232 soros port monitor
VBRUN100 — Visual Basic runtime modul
VBSYSINF — Rendszerinformációk
WBAR18 — Vonalkódprogram Windows alatt
WBAUTO11 — WallBaster-kiegészítés
WCLOCK — Óra

Lemezszám: 566

Név: Windows-segédprogramok/18

Leírás: 11 program Windows alá
WINCLI10 — Utasításinterpreter
WINCRON — Programindító adott időre
WINFO10 — Rendszerinformáció
WINPLAY — Polifonikus zene
WINQWK10 — Gyors üzenetolvasó
WINRUN10 — Alkalmazás RunUp segédprogram
WINTIDE — Óra
WINUNZIP — ZIP állománykicsomagoló
WLNCH21 — Konfigurálható programmenedzser
WUNZP107 — ZIP állomány kicsomagoló
WYW020 — Amíg távol volt — telefonüzenetek

mindez bombabiztos futtatható kódot eredményez, ami szinte szükségtelenné teszi a programtesztelési fázist.

Konkurens programok futtatása

A hatékony programírást szolgálja a TSC overlaymenedzsere, amely lehetővé teszi a DOS 640 K-s korlátjának megkerülését, s így már maximum 16 Mbájtnyi DOS programot is gyárthatunk. (A TSC belső függvényeinek és a teljes, szabványos ANSI C könyvtárának forráskódja egy különálló, TSC Library Source Code Kit nevű csomagban megvásárolható. Ebben a teljes overlaymenedzser forráskódját is mellékelték!)

A TSC példa nélkül álló szolgáltatása a többszörös ún. threadek támogatása. A TS alatt (MS-DOS vagy OS/2 operációs rendszerben) egyszerre maximum 32 önálló folyamat vezérelhető egymással konkurens módon (a TechKit és SourceKit segítségével 32-nél is több). A főprogramból leágazó fonalak (threads) szubrutinhívásokon keresztül működnek együtt a beépített időosztásos ütemező (time-slicing scheduler) vezényletével.

A TopSpeed Pascal

A „felesküdtek" nem fognak retirálni...

A TopSpeed Pascal (a továbbiakban TSP) együttműködik a TS objektumorientált (OOP) fejlesztői környezetével, és mint ilyen az 1990-es évek első komplett OOP-alkalmazása.

A profi programozók számára a TSP számos olyan forradalmian új technikát nyújt, mely a maga nemében egyedülálló. A világon elsőnek teremti meg a TSP azt a lehetőséget Pascalban, hogy ne csak MS-DOS, hanem OS/2 és Windows 3.0 alá is tudjunk alkalmazásokat írni.

A többszörös öröklés (multiple inheritance), a virtuális és statikus eljárások, az automatikus objektuminicializáció az igazi magja egy mai hatékony objektumorientált rendszernek. A biztonságos futásidejű típusellenőrzés, az elkülönülő fordítási menetek (implementáció és interfész), valamint a TopSpeed intelligens linkelési funkciója teremtik meg a lehetőségét, hogy valóban megbízható és szuperhatékony legyen a kód és maga a generálása.

A TSP-ben az osztályok akár egynél több alaposztály tulajdonságait is örökölhetik. Ez pontosan azt jelenti, hogy maximálisan fazonra szabott osztályokat lehet készíteni minimális energiabefektetéssel és a lehető legrövidebb időn belül. A TSP egyaránt támogatja a virtuális és statikus eljárásokat, ami nagymértékben megnöveli a rugalmasságot.

A TSP adatbiztos konverziót valósít meg az egyes objektumok között, könnyen elérhetővé téve így az egyik osztály funkcionalitását a másikból. A TSM2 beépített öröklésének köszönhetően olyan új osztályok létrehozása, amelyek egy már meglévőn alapulnak, jelentősen leegyszerűsödik. (Ide tartozik, hogy mivel valamennyi TopSpeed objektumorientált nyelvjárás kompatibilis egymással, így nyugodtan programozhatunk többnyelvű környezetben is: Modula-2-ben, Pascalban vagy C++-ban. A TS Smart Linking technológiája és a SmartClass linkelési eljárás mindig hatásos — lásd e rovatban az e cikket megelőző TS-körű írásokat. Azokból egyéb, az összes TS nyelvre/-környezetre vonatkozó tulajdonságok is megismerhetők.)

A memóriamodellek variálása

Ha a futtatható állomány esetében a gyorsaság és a minél kisebb méret a szempont, maradjunk a small modellnél, mivel ez csak szegmensen belüli hívásokat és pointereket használ. Ha nagyméretű adatstruktúrákkal van dolgunk, használjunk far pointereket. Ha a programkódunk mérete meghaladja a 64 K-t, nyilvánvalóan olyan modellt kell választanunk, amely támogatja a szegmensek közötti függvényhívásokat is (far calls). A táblázatban megnézhetjük, milyen lehetőségekkel számolhatunk:

Modell	Max. adat	Max. kód	Calls & Rets	Data Pointers
Small	64 K	64 K	Near	Near
Compact	1 MB	64 K	Near	Far
Medium	64 K	1 MB	Far	Near
Large	1 MB	1 MB	Far	Far
Xlarge	16 MB	16 MB	Far	Far (OS/2 esetén)

Fejlesztés TSP-ben

Más Pascal-implementációkból egyszerűen át lehet hozni a TSP-be a forrásprogramokat. Egy TopSpeed segédprogram segít átkonvertálni a Borland Turbo Pascalban írott forráskódot. A legtöbb Turbo Object is könnyedén átvehető, csupán néhány helyen kell belenyúlnia az avatott kezű programozónak. A többnyelvű környezet megteremti annak a lehetőségét is, hogy akár Modula-2-ben készítsük el a programozói team (csapat) Pascal, C++ és C rutinkönyvtárait és viszont. Az összes TS nyelv végső soron ugyanazt a kódgenerátort használja, így bármelyik TS nyelv esetében ugyanazt az elit minőségű, gyors és tömör futtatható kódot kapjuk.

Át sem kell térni az OS/2 operációs rendszerre ahhoz, hogy multitasking környezetben futtathassunk Pascal programokat. Mindez megtehető MS-DOS-ban is, ugyanis a TSP támogatja a multithreading szoftvertechnológiát. A „thread" egy olyan szubrutin, mely a mindenkori főprogrammal párhuzamosan fut. Időosztásos üzemmódban dolgozhatunk. Egyszerre maximum 32 thread működhet együtt, de a TopSpeed TechKit és SourceKit felhasználásával ez a korlát jelentősen megnövelhető, a gyakorlati határ csupán a rendelkezésre álló memória mérete.

Tulajdonságok, képességek, árak

— ISO 7185 level 1 szabvány szerinti Pascal.
— Borland Turbo Pascal 5.x és 6.0 konverter TSP kódra, a hatékonysága jobb, mint 90%.
— Objektumorientált programozástechnikai bővítések.
— Regiszterpáron keresztüli paraméterátadási konvenció.
— ISO-szabvány szerinti tömbök (akár 16 MB méretben is!).
— Dinamikus szöveges függvények.
— Elkülönített fordítási menet az implementáció és az interfész között.
— Teljes futásidejű nyomkövetés és hibakeresés.
— Feltételes compilerutasítások egyszerre több programváltozat kezeléséhez.
— Teljesen ISO Pascal-kompatibilis library.
— Inline assembler.
— Abszolút változók kezelése, melyeket előre megadott helyre tesz a memóriában a betöltéskor a program.
— Egyszerre 32 konkurens program futhat, dead-lock-elkerülő algoritmus.
— Karakterbázisú beépített ablaktechnika.
— 16 bites short és standard far pointerek.
— TopSpeed VID (Virtual Interactive Debugger) vagy Microsoft CodeView használata hibakereséshez.
— 80x87-es koprocesszor támogatása.
— Automatikus I/O prioritáskezelés az OS/2-es változatban.
— Standard Microsoft .OBJ kódkompatibilitás.
— Az OS/2 változat real és protected módú programot is létre tud hozni.
— További fejlesztői libraryk: File I/O, string, DOS/BIOS és a Windows-funkciók közvetlen eléréséhez.

Árak az Alaplap Postán keresztül:
TS Pascal 1.0 Compiler 12 000 Ft + ÁFA, TS Pascal Standard Pack (Compiler + TS Environment) 24 000 Ft + ÁFA, TS Pascal Professional Pack (Standard + TechKit) 39 000 Ft + ÁFA, TS Pascal Library Source Kit 18 000 Ft + ÁFA.
Megjegyzés: az OS/2-es változat árai megegyeznek a fentiekkel.

A TSP Library Source Code Kit

A TSP belső függvényeinek forráskódja egy különálló csomagban megvásárolható. A teljes TSP rutinkönyvtár forráskódját kézhez kapjuk, beleértve a 80x87-es emulátort és a start-up függvényeket is. A lemezen — gazdagon ellátva magyarázatokkal — minden szoftvererőforráshoz hozzáférhetünk, ráadásul még számos OOP-mintaprogramot is mellékeltek. (A Library Source Kit-hez a TS TechKit megléte szükséges.)

— ISO + TopSpeed Library:
— Console & file I/O, redirection.
— Memóriamenedzsment, lebegőpontos aritmetika!
— Adattípus-konverziók, stringfüggvények.
— Memóriablokk-kezelés.
— OOP Library.

A TopSpeed Modula-2 v3.02

„A programozás legmagasabb csúcsa"

A TopSpeed nagyon gyorsan kinőtte magát mint a Modula-2 nyelv legelismertebb megvalósítása MS-DOS és OS/2 alatt. A hatékony, de könnyen elsajátítható nyelv, a Modula-2 nemcsak növeli a profi programozó termelékenységét és eredményességét, hanem egy nagyságrenddel megemeli a végső program színvonalát. (A nyelv lehetőségeit gazdagon illusztráló sorozatunk a Programozástechnika rovatban novemberben tovább folytatódik — ezúttal a TS-csokor részeként a Modula mint szoftvertermék az érdekes.)

Niklaus Wirth, a Pascal programozási nyelv szülőatyja a Modula-2 nyelvet úgy tervezte, hogy az a szigorúan adattípusokra épülő, strukturált magas szintű nyelvek minden előnyét hatványozottan egyesítse, így megközelítse a C vagy az assembler hatékonyságát. A Jensen & Partners International (JPI) átvette ezeket az alapdefiníciókat, és egyéb kibővítések széles tárházával tette teljessé a Modula-2-t (továbbiakban TSM2).

Nem csoda ezek után, hogy a TSM2-t „a programozás legmagasabb csúcsának" titulálta annak idején a PC Week 1991. januári száma.

A JPI TSM2 implementációja az objektumorientált programozáson keresztül az egyszerre több magas szintű nyelven történő fejlesztésig rengeteg hasznos szolgáltatást nyújt.

Élen az OOP-ben

A TSM2 valóban a számítógépes szoftvertechnológia legmagasabb dobogójára érdemes. A többszörös öröklés (multiple inheritance), a virtuális és statikus eljárások, az automatikus objektuminicializáció az igazi magja egy mai hatékony objektumorientált rendszernek. Biztonságos futásidejű típusellenőrzés, elkülönített fordítási menetek (implementáció és interfész) a TopSpeed intelligens linkelési funkciójával párosítva teremtik meg a lehetőségét valóban megbízható és szuperhatékony, gyors kód generálásának. A TSM2-ben az osztályok több mint egy alaposztály tulajdonságait is örökölhetik. Ez pontosan azt jelenti, hogy maximálisan fazonra szabott osztályokat lehet készíteni minimális energiabefektetéssel, és a lehető legrövidebb időn belül. A TSM2 egyaránt támogatja a virtuális és statikus eljárásokat, ami nagymértékben megnöveli a rugalmasságot.

A TSM2 adatbiztos konverziót valósít meg az egyes objektumok között, könnyen elérhetővé téve így az egyik osztály funkcionalitását a másikból. A TSM2 beépített öröklésének köszönhetően olyan új osztályok létrehozása, melyek egy már meglévőn alapulnak, jelentősen leegyszerűsödtek.

A TopSpeed minden egyes objektumorientált nyelvjárása kompatibilis egymással, így nyugodtan programozhatunk többnyelvű környezetben is: Modula-2-ben, Pascalban vagy C++-ban.

Az összes TopSpeed nyelv képes kevert memóriamodellen alapuló programozásra (mixed model programming), de maga a TS-compiler is számos előre definiált memóriamodell-sablonnal rendelkezik. A függvényhívási konvenciók komplett kezelése és a TS átfogó pragmarendszere (a fordítás menetét vezérlő, azt elágaztató vezér-

lőutasítások) nagyban támogatja a fenti programtechnikákat.

Az új pointertípus

A TSM2 legutolsó változata egy új pointertípust vezetett be: a virtuális pointert. Ez a típus, ha nem hivatkoznak rá, aktivizál egy felhasználó definiálta függvényt. Ez a függvény adhatja meg a pointer valódi értékét. A virtuális pointerek révén folyamatos (non-fragmenting) heapet, a nem használt adatterületek felszabadítását (garbage collection), virtuális memóriakezelést, vagy más bonyolult feladatokat könnyedén lehet megoldani.

A „thread"-ek, azaz konkurens programok futtatása

A TSM2 is támogatja a többszörös threadeket. A TS alatt MS-DOS és OS/2 alatt egyszerre maximum 32 önálló folyamat vezérelhető egymással konkurens módon (a TechKit és SourceKit segítségével 32-nél is több).

A főprogramból leágazó fonalak (threads) szubrutinhívásokon keresztül működnek együtt a beépített időosztásos ütemező (time-slicing scheduler) vezényletével.

Tulajdonságok, képességek, árak

— Objektumorientált programozástechnikai bővítések: multiple inheritance, automatic constructors.
— Speciális compilervezérlő utasítások (pragmák): inicializáló kód generálásának tiltása, a változó hosszúságú rekordok nem statikusak (volatile), azonos mezőnevek használatának megengedése a variant rekordokban, a számított típusok (enumerated types) méretének megadása, az eljárások nyomkövetésének engedélyezése, illetve tiltása.
— Típuskorlátozás (type coercion).
— 80x87-es koprocesszor támogatása.
— Paraméterátadás regiszterpárokon keresztül, csökkentett stackigénybevétel. Compiler pragmákkal választható.
— Szegmensen belüli pointerek (short based pointers).
— A TSM2 Library tartalmazza a teljes szoftverfelületet az MS-DOS, a BIOS és az egérhandlerek felé, plusz karakterbázisú ablakkezelő rendszert is biztosít.
— Inline gépi kódú sorok (assembler kód a normál forrássorok között).
— LIM EMS memóriabővítés támogatása, oda adatterület feltöltése.
— Többszörös, kevert memóriamodell keverése.
— Smart linking.
— Feltételes fordítási lehetőség, mely lehetővé teszi különböző eltérő változatok és tesztverziók kezelését.
— Teljes futásidejű hibakezelés.
— Windows 3.0 kód generálása.
— Virtuális pointerek.
— Az OS/2 változat real és protected módú programot is létre tud hozni.
— Az OS/2 Presentation Manager támogatása.

Árak az Alaplap Postán keresztül:
TS Modula-2 Compiler 12 000 Ft + ÁFA
TS Modula-2 Standard Pack (Compiler + TS Environment) 24 000 Ft + ÁFA
TS Modula-2 Professional Pack (Standard + TechKit) 36 000 Ft + ÁFA
TS Modula-2 Library Source Kit 12 000 Ft + ÁFA

A TSM2 belső függvényeinek forráskódja egy különálló csomagban (TSM2 Library Source Code Kit) megvásárolható. Ugyanazt a forráskódot kapjuk DOS és OS/2-höz egyaránt, mely Modula-2 és assembler nyelven íródott.

A Library Source Kit (a TS TechKit szükséges) tartalma:

— TopSpeed Modula-2 Library:
— IO, BiosIO, FIO, FIOR, Graph, Lib, MATHLIB, MsMouse, Proces, ShtHeap, Storage, Str, SYSTEM, Window;
— PIM-kompatibilis core library;
— Terminal, FileSystem, InOut, RealInOut, MathLib0.

Rogue Wave C++ Class Library (RWCL)

Plusz-kettősök

A JPI a TopSpeed C++ fejlesztői környezet kiegészítéseként kiadta a Rogue Wave C++ rutinkönyvtárait is. A termékben két önálló rutincsomag található, a Tools.h++ és a Math.h++ nevű könyvtár.

A RWCL-ben található önálló osztályok anélkül is felhasználhatók, hogy a programozónak teljes mélységig értenie kellene az egyes osztályok működését önmagukban.

Az objektumorientált programozástechnika egyik alapkövét, a többszörös öröklést felhasználva saját osztályainkat pillanatok alatt származtathatjuk az RWCL gyári moduljaiból.

Tools.h++

— Gyűjtőosztályok (Collection classes). A Smalltalk-80 alapján készült programozói környezetet biztosít: Set, Bag, Ordered Collection, Sorted Collection, Dictionary, Stack, Queue stb.

— Általános osztályok: singly & doubly-linked lists, stacks & queues, paraméterezhető típusok.

— B-Tree algoritmust használó lemezkezelő. Hatékony, kulccsal elérhető rekordkezelés a háttértárolón.

— File Space Manager. Helyet allokál és deallokál egy lemezes állományban.

— String- és karakterkezelő osztályok. Operátorok és függvények gazdag kínálata (konkatenáció, összehasonlítás, indexelés, alsó-felső határtúllépés ellenőrzése). SubString osztály.

— Idő- és dátumkezelő osztályok.

Math.h++

— Vektorok és mátrixok. Tetszés szerinti adattípusra: double, complex, int, float, signed & unsigned char. Komplett matematikai függvénygyűjtemény túlcsordulás-kezeléssel, gyors adattárolási és visszakeresési eljárásokkal.

— Slice & Pick operátorok (részmátrixok gyors, konform kezelése).

— Bounds Checking (ellenőrzés indextúllépés megelőzésére vektorok és mátrixok körében.

— Komplex számok.

— Gyors Fourier-transzformáció. Egy- és kétdimenziós FFT (Fast Fourier Transforms), szinusz és koszinusz transzformációk.

— Lineáris algebra. Mátrix invertálása, transzponálása, lineáris egyenletrendszerek megoldása, determinánskeresés.

— Statisztika. Statisztikai és valószínűségi függvények és eloszlások.

— Hisztogramok osztálya. Lineáris regresszió.

— Véletlenszám-generálás különböző eloszlásokhoz (normál, Poisson, Lorentz és a többi...)

— A teljes C++ forráskódot is mellékelik!

Ára az Alaplap Postán keresztül: TopSpeed Rogue Wave C++ Class Library 12 000 Ft + ÁFA.

HC Hun Comp

Elektronikai Szolgáltató és Kereskedelmi Kft.
1116 Budapest, Mohai út 37. • Tel. & Fax: 185-4186

Teljes gépösszeállítások

1. AT 286 CPU 16 MHz, 1 MB RAM, 1,2 MB FDD, HDC/FDC, 101 gombos bill.	23.000,–
2. AT 386SX CPU 25 MHz, mint az 1. tételnél	29.000,–
3. AT 386 CPU 40 MHz, mint az 1. tételnél	28.000,–
4. AT 486 CPU 33 MHz, mint az 1. tételnél	35.000,–
5. AT 486 CPU 50 MHz, mint az 1. tételnél	38.000,–

Monitorok a gépekhez

14" monokróm + kártya	9.000,–
14" monokróm VGA monitor	15.000,–
VGA, 1024 monitor + kártya	29.000,–

Winchesterfelárak

40 MB (AT BUS)	17.000,–
80 MB (AT BUS)	24.500,–
124 MB (AT BUS, 15 ms)	29.000,–
212 MB (AT BUS, SEAGATE)	49.900,–
SIMM 1 MB RAM	2.500,–

EPSON, HP, CANON ÉS STAR TERMÉKEK TELJES VÁLASZTÉKA

NOVELL ÉS UNIX TERMINÁLOK, HÁLÓZATOK IGÉNY SZERINT !

Az árak az ÁFA-t nem tartalmazzák.

GARANCIA: 1 ÉV

INFORMÁCIÓKÉRÉS: 32 ▲

Ha kinőtte a CLIPPER-t,

és nem akar csöbörből vödörbe esni,

akkor az ideális megoldás:

SYBASE.

Már Novell NetWare változatban is!

Kliens/szerver architektúra, 3GL+4GL, osztott adatbáziskezelés, programozható szerver, magas teljesítmény, automatikus lekérdezés-optimalizálás, opcionális Windows felület. Heterogén hálózatok (Sun, VAX, HP, IBM, DG, Sequent stb.), teljes életciklus támogatás, CASE, ...

Axis

Számítástechnikai Kft.

8001 Székesfehérvár, Pf.98 (Gyümölcs u. 4.)

☎ (22) 27-631 ❖ Fax: (22) 27-630

INFORMÁCIÓKÉRÉS: 03 ▲

BLAISE INFORMATION BIS SECURITY

Tel./Fax: 251-5873, 252-1500
1145 Budapest, Róna u. 127/B.

ADATVÉDELEM Kft.

A DIGI... család

DigiStock
Értékpapír-archiválás

DigiDat
Napi iratkezelő és archiváló rendsze

DigiFile
Általános dokumentumkezelés és archiválás

DigiSign
Digitalizált aláírásminták adatbázis-rendszere

CANON NAVIGÁTOR. A holnap irodája ma!
– PC XT, fax, telefon, üzenetrögzítő
– Szkenner, másoló, nyomtató (2-féle)
– Különféle alkalmazói szoftverek
Mindez egy teljes rendszerben!

DYCAM CAMERA. A számítógép szeme!
Digitális hordozható kamera

BIS SEC 1.0
Univerzális PC adatvédelmi kártya

INFORMÁCIÓKÉRÉS: 05 ▲

KIÁRUSÍTÁS!

Az Everlead Kft. kiárusítja a raktáron lévő PC-alkatrészkészletét.

Alaplapok
Kártyák
Tápegységek
Dobozok
DR. DOS 5.0
DR. DOS 6.0

Cím: Budapest XIV., Korong u. 35.
Tel.: 252-6969 • Tel./Fax: 252-6475

INFORMÁCIÓKÉRÉS: 22 ▲

Itt a pénz, hol a pénz? II.

Nézd a mágneskártyád, ott van egy szám!

Az adatbank és az azonosító rendszer felépítésének tárgyalásánál a következő régi kérdés tisztázandó: a farok csóválja a kutyát, avagy fordítva? A számítástechnikai megoldások adják egy banki üzletág kereteit, vagy a számítástechnikai eszközöknek, megoldásoknak kell szolgálniuk az üzletág mögött álló ügyfeleket?

Számítástechnikai szempontból a készpénzkímélő banki rendszer megoldása az alábbi lehet:

Vásároljunk egy jó nagy kapacitású számítógépet. Kössük össze az ország számos megyéjének számítógépeivel. Ezek a központi gépek az ügyféllel kapcsolatban álló fiókok termináljaival állnak kapcsolatban. Kössük össze továbbá a készpénzkezelés céljára alkalmazott berendezésekkel. Telepítsünk a rendszerre néhány átvitelt irányító, vezérlő szoftvert, és sok-sok apró szoftvert, melyek a hierarchia minden szintjén megfelelő szolgáltatást biztosítanak.

Informatikailag azonban a modell más alapállásból is felépíthető:

A szolgáltatást igénylők többsége a lakóhelyéhez legközelebbi pénzintézetnél vagy annak fiókjánál veszi igénybe a szolgáltatásokat. (Nem várható, hogy egy adott időszakban minden ügyfél egyszerre átköltözik mondjuk Heves megyéből Somogy megyébe egy kicsit költekezni.) Magától értetődik, hogy a legrészletesebb információigényt a megyei központoknak kell kiszolgálniuk. Az ügyfél leggyakrabban a számlavezető bankjának fiókjával találkozik, ott reklamál, érdeklődik. Ehhez komoly számítástechnikai felszerelésre lesz szükség.

Míg az előzőekben tárgyalt koncepció szerint e megyei központoknak az országos központ gépeit is ki kell szolgálniuk, valószínűleg redundáns módon, lokálisan is az adatok birtokában kell lenniük.

Természetesen választhatunk, hogy vagy az országos távadatátviteli hálózatot vesszük igénybe (ami költséges szolgáltatás), vagy pedig csakis akkor igényeljük az ilyen kapcsolatok kiépítését, ha a szolgáltatás igénypontjai saját rendszerünk határain túlmutatnak. Ez utóbbi egyszerűbb, olcsóbb, de nem rosszabb modell.

Gondoljuk tovább!

Tételezzük fel, hogy a megyei központok a kapcsolattartást műholdon keresztül bonyolítják le, topológiai korlátozás nélkül. A távkapcsoláshoz azonban akkor és csakis akkor kell fordulni, ha a rendszerhatáron túllépő tranzakció jelentkezik. Óvatos becslés alapján a lakossági összforgalom 25-30%-a igényli ezt a kapcsolattartást.

A teljes koncepcióban a hierarchia legalsó szintjén álló elfogadóhelyek elfogadói végpontok, pénzkiadó automaták, bankablaki kifizetőhelyek, vagy vezetéken, illetve rádiómodemen keresztül kapcsolódnak egy-egy helyi bank központi számítógépéhez. Egy-egy megyében ezek a számítógépek egymással vagy vezetékes, vagy rádiómodemes kapcsolatban állnak, melyeken egyéb banki tranzakciók is forgalmazódnak. A megyei központok a másik megyei központot műholdon keresztül érik el. Ily módon a kapcsolatok mobilak, és az országos hálózatokat csak akkor és addig veszik igénybe, ameddig és amikor erre feltétlenül szükségük van.

„Nyomozati anyag"

Ha a fenti koncepciót vesszük alapul, egy-egy tranzakció eredeti forráshelyére magából a kártyaszámból kell következtetnünk. A kártyaszám 16 számjegy, mely a nemzetközi kártyakibocsátás azonosítószámát tartalmazza. Ebben benne foglaltatik a kibocsátó ország kódja, a kibocsátó szerv egyedi azonosítója.

A kód megszerzésére nem minden bank pályázhat megfelelő eséllyel. Elképzelhető olyan laza egyesülés, melyben az egyenként esélytelen pályázók a cél érdekében egységesen lépnek fel. A takarékszövetkezeti bankhálózat mint országos szövetség egységesen veheti a nemzetközi jegyzést igénybe, azonban bankjai önállóságukat továbbra is megtartva üzemelhetnek. Az OTP-nek viszont egy-egy megyei központja lehet önálló kibocsátó. (A megyei központ azonban nem feltétlenül Magyarország megyehálózata szerint értendő.)

A fentiek alapján a következő azonosítószám-igény merül fel: 3 számjegyes a kibocsátó fiók kódja; 1 számjegy a kártya típuskódja.

A jelen gyakorlatra három kártyatípus jellemző:

— JUNIOR-kártya (kizárólag készpénzfelvételre alkalmas).

— Bankkártya (készpénzfelvételre és vásárlásra alkalmas; természetesen pontosan szabályozott, azaz limitszintekhez kötött érvényességi körrel).

— Vállalkozói kártya (igen hasonlít a bankkártyához, csak a limitszintek lényegesen kedvezőbbek).

Még sok más kártyatípus van, szerepük bizonyára a jövőben alakul ki.

A 16 darab számjeggyel, melyet a mágnescsíkon rögzítenek, a fentiekben felvázolt minden követelmény hosszabb távon is teljesíthető. Mindenesetre 5 számjegy hosszban a kártya sorozatszáma szerepel. (Ezt a kibocsátó számítógépes rendszere generálja.) 1 számjegy egy ellenőrző szám, amely valószínűsíti, hogy a számsorozat egyáltalán kártyaszám lehet-e. Ennek algoritmusát érthető okok miatt nem közöljük. Az azonosító további részeit a következő megfontolások eredményezik.

Kiváltjuk a kártyát

A természetes vagy jogi személy elhatározza, hogy a fenti szolgáltatást igénybe veszi, tehát kiválasztja azt a kibocsátót, aki a számára a legkedvezőbb opciókat nyújtja, majd szerződést köt vele. A bank és az igénylő közötti

szerződésben pontosan feltüntetik a nyitás dátumát, az igénylő pontos címét, egyéb adatait, a szerződés érvényességi idejét, az induláskor esetleg igényelt hitel összegét.

Ez utóbbi fontos momentum. Elképzelhető, hogy a kártyahasználó a vásárlások során a betét összegét túllépi. Ezt a pénzintézmények különbözőképpen viselik el, mindenesetre a túllépés kiegyenlítéséig büntetőkamatot számítanak fel. Ha szükséges, a standard banki hiteligénylésnek megfelelően egyezséget köthet a felhasználó a bankkal, hogy ilyen esetre tartson hitelkeretet. Ekkor sima hitelügyletben a vásárlás lebonyolítható. Ha a hitelkeretet is túllépte, csak akkor kerül a büntető hitelágba az ügyfél. (Online rendszerben ez nem következhet be: vagy van fedezet, vagy nincs. Persze ezt a bank üzletpolitikai okból feloldhatja, akár ügyfelenként is.)

A szerződés alapján kártya adható ki. Egy életszerű ügy: a NONAME Kft. vezetője nyit egy vállakozói kártyát, hogy a szerződés alapján vezetett pénzéhez bármikor hozzáférjen. Három anyagbeszerzője részére is vállalkozói kártyát kér, mert nem kívánja a készpénzszállítás kockázatát vállalni. Felnőtt családtagjainak azonban bankkártyát nyit a napi költségeikhez. Gyermekének az iskolai kiadások fedezetére JUNIOR-kártyát vált. Tevékenysége határon túli szállítmányozásra is feljogosítja. A vámilleték lerovására, a vámolási procedúra egyszerűsítésére vámkártyát nyit mindkét gépkocsivezetője részére. (A határátlépéskor az illetékes vámszervek felbecsülik a várható vámilletéket, ezt átutalják a konkrét vámhivatali szervnek, ahol a részletes vámolás megtörténik. A vámilleték részletes megállapítását követően az összeget aztán leemelik a számláról, avagy a többletet visszautalják.)

Teheti ezt a NONAME Kft. vezetője, mert a bank minden hónapban kártyánkénti kimutatásban értesíti a pénzforgalomról. A kimutatás bizonyító erejű dokumentum, a különböző forgalmakból leválogatható, mi tartozik a vállalati kosárba, és mi az egyénibe. Természetesen a mindenkori egyenlegről a bank személyes kérésre a legteljesebb mértékben informálja, azonban csak és kizárólag a szerződéskötő személyt. (Végeredményben a bank csak vele áll kapcsolatban!)

Az ugye kézenfekvő lehetőség, hogy a szerződéskötő felhatalmazza a bankot: az általa igényelt egyéb kártyákkal az általa megnevezett személyeknek is álljon a rendelkezésére. Ha a szerződéskötő ilyen jogot adhat, el is veheti. Ezt a kártyarendszer kezelésimód-váltásnak nevezi.

Lehet egy szerződésnek akármennyi aktív — forgalmazható — állapotú kártyája, minimálisan két kezelési mód és vezérlési mód alkalmazandó: egy vagy több kártyaforgalom aktiválása vagy inaktiválása biztosítandó; az egész szerződés érvénytelenítése (ez utóbbi egyben minden kártya megszüntetését is jelenti).

Az érvénytelenített kártyákat vagy a szerződéskötő szolgáltatja vissza a banknak, vagy csak rendelkezik róla. (Ez olyan esetben fordulhat elő, ha a kártyahasználókat a szerződéskötő nem éri el.) A forgalmazhatatlan kártyákat bármely pénzkezelő automata automatikusan bevonja. Természetesen bizonyos bírósági és egyéb államigazgatási eljárások részben vagy egészben, végérvényesen vagy időszakosan a kártyaforgalmat szüneteltethetik. A kódoknak ezekre az esetekre is vonatkozniuk kell.

A szoftver dolga

Általában egy-egy szoftverrendszer addig hatásos, amíg a reálszféra változásai a szoftvernek részben vagy egészben merev rendszerét szét nem feszítik. (Ki ne ismerne olyan rendszereket, melyek az elkészülésük pillanatában már elavultak voltak?!) Ahhoz, hogy a nem kis szellemi és anyagi befektetéssel elkészült szoftverterméket hosszabb távon is használhassunk, meg kell vizsgálnunk, hogy a szoftvernek milyen valós vagy vélhetően bekövetkező helyzetekben kell helytállnia. Ezt a mai gyakorlatban paraméterezésekkel érjük el. Az, hogy mit kell, és miként kell paraméterezni, azaz a működés mikor, miként történjék, nem a programozó döntése. (Természetesen a hardverismeret a programozókat ezen a területen előnyben részesíti. De csak itt!) Ehhez az adott terület teljes ismeretén túl az információszervezés, elemzés, modellkészítés ágazataiban jártas szervező szakemberek kellenek. (A tényszerű elemzés alapján kerül sor többek között annak megvitatására is, hogy az adott feladatot egyáltalán számítógéppel ésszerű-e megoldani.)

Az eddig elmondottak alapján a témakörhöz tartozó néhány szempontot sorolunk fel. Ezeknek a rendszerben érvényesülniük kell, illetve ezek olyan dolgok, amelyeknek a figyelése/kezelése a rendszer alfája és ómegája.

Állapotjelző. Mutatja, hogy egy-egy rekord fizikailag milyen állapotban van (például a SZERVIZ analóg valamilyen sérüléssel).

Feldolgozási állapotjelző. Egy-egy rekord a feldolgozhatóság szempontjából milyen állapotban van. (Feldolgozásra vár, feldolgozásban részt vett, részfeldolgozott stb.) Az esetleges újbóli indulásnál honnan kell a tranzakciókat folytatni?

Rekordműveleti állapotjelző. Egy többfájlos feldolgozásnál bármikor megszakíthatják (hardverhiba!) a rekordok felírását, módosítását. Vajon minden sikerült?

Személyi azonosító. A műveleteket személy szerint ki, mikor végezte? Más, a rendszert karbantartó személyzet kezeli, vagy a rendszerfejlesztő, illetve felhasználó személyzet.

Főkönyv. Milyen főkönyvi számokat használunk, és azokat miként? (Aktuális a kérdés az új számlarend beléptetésével.)

Mozgások. A rendszer milyen mozgásokat kezel, ezeket ki használhatja?

Kondíciók. Milyen kamatértékekkel számoltak, milyen költségszámításokat alkalmaztak?

Algoritmusok. A kondíciók százalékszámítások, fix összegek, növelő/csökkentő típusúak, sávos/progresszív számításúak-e, s mindezek időszakonkéntiek, rendszerben globálisak, adott szerződésre lokálisak lehetnek.

Printer. Vajon milyen típusú a nyomtató, mi a vezérlőkarakter-sorozata, melyik porton van, lokális használatú-e vagy hálózati?

Archív. Milyen adathordozóra archiválunk, mivel és mikor?

Modem. Milyen modem van, hol van, mi vezérli?

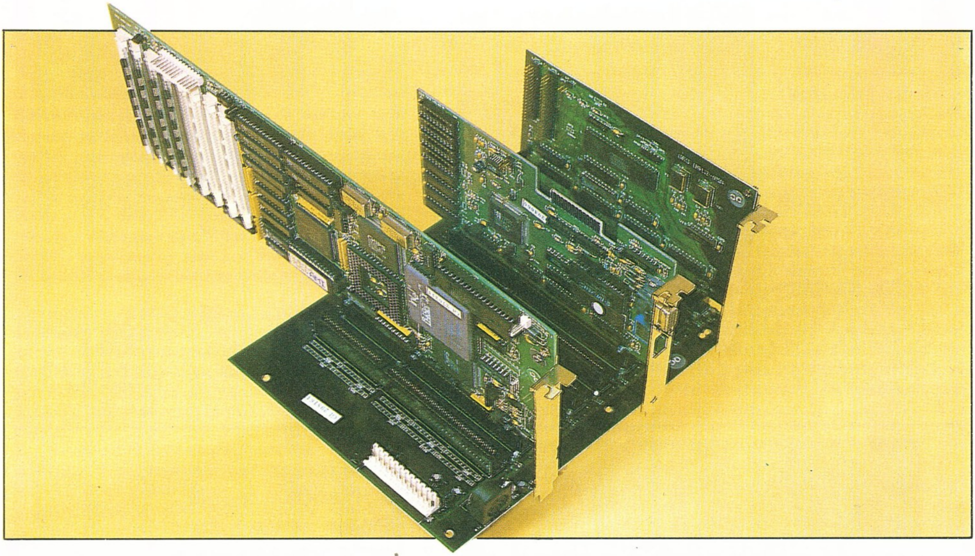
Hálózat. Egyfelhasználós/többfelhasználós? DOS? Unix? OS/2?

Starcz Andor

Mindent egy kártyára!

MULTIflex (MF Series) SYSTEM

SZILICIUM ELEKTRONIKA Kft.
BP. 1065 NAGYMEZŐ U. 66. • TEL.: 1321-912 • FAX: 1327-572



A MULTIflex rendszer új szemléletet és új eszköztárat ad számítógépünk korszerűségének megőrzéséhez. A szokásos alaplap helyén csak üres foglalat van, s egy könnyen cserélhető kártya hordozza a processzort, a RAM-ot, a BIOS-t és az akkumulátort. A többi szokásos kártyával az egységesített AMI BIOS teremti meg az együttműködést.

A CPU kártyára biztosított 2 év garancia önmagában is jelzi a rendszer megbízhatóságát és minőségét. A kártya japán technológiával készül, többféle felépítésben:

— AT 286, 16 MHz, 1 MB RAM
— AT 386SX, 20 MHz, 2 MB RAM, 128 K cache
— AT 386DX, 33 MHz, 4 MB RAM, 128 K cache
— AT 486SX, 20 MHz, 4 MB RAM, 128 K cache
— AT 486DX, 33 MHz, 4 MB RAM, 128 K cache
— AT 486DX, 2—50 MHz, 8 MB RAM, 256 K cache (DUAL CLOCK)

Számítógépünk megbízhatóságát szolgálja a teljes Fujitsu merevlemez-választék, a WD Paradise VGA kártya, a 6-féle monitor... és a működtetéshez szükséges összes tartozék.

A dealereket és a viszonteladókat is várja
a MULTIflex berendezések kizárólagos hazai forgalmazója:

INFORMÁCIÓKÉRÉS: 49 ▸

Interfész az építész agya és a szoftver között

Emlékkápolna

Egy meghatározó gyermekkori emlék és egy zseniális építészeti elképzelés együttesen arra ihlette építészdinasztiából származó szerzőnket, hogy elkészítse a Don-kanyar Emlékkápolna számítógépes tervét. Az egyedülálló, nagyszabású látvány- és kiviteli terv kommentálása így természetesen nem mentes — a pozitív értelemben vett — szubjektív felhangtól.

Emlékképek...

Egy építész számára valóságos kíhívást jelent, amikor megtudja, hogy a Don-kanyar Emlékkápolna építésére alakult alapítvány tervpályázatot hirdet. Egyből eszébe jut, hogy nagyapjának — mint oly sok más magyar család idős tagjának — a Don-kanyar a borzalmakat, a kínszenvedést és a meghurcolást jelentette. A nagyapa még szerencsére élt 35 évet, de a Don-kanyar kitörölhetetlen emlékeket hagyott nemcsak az ő, hanem valamennyi családtagjában. Így az építész unoka — tisztelegve nagyapja emléke előtt is — munkához lát.

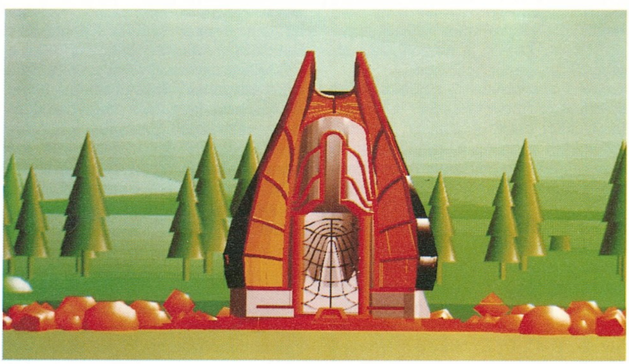
A Krisztus-lepel

A Krisztus-lepel motívum jelenti az alapját annak az építészeti elképzelésnek, amelyet a lelkes alapítványtevők a pákozdi emlékmű közelében építenének fel. Bizonyára már sokan láttak olyan Jézus-ábrázolást, ahol Krisztus kicsit nyitott testtartásban emeli fel mindkét kezét, hogy megáldja népét. Eközben az általa viselt lepel egy szép térbeli felületet ír le.

Az alapítványtevők elvárásai olyan komplexitást igényeltek, amire eddig még nem volt példa Magyarországon: a térbeli modell segítségével animációt, látványképeket, engedélyezési és kiviteli terveket kellett készíteni. A feladatot tovább bonyolította, illetve szépítette az az alaptestekből le nem írható összetett térbeli felület, amely láttán több építész kolléga hátán — elég látványosan — futkosott a hideg.

Térbeli kihívások

Egy „klasszikus" építész nem 1000 oldalas dolgozatban írja le, hogy milyennek képzeli a leendő épületet, hanem műszaki rajzzal valósítja meg elképzeléseit.



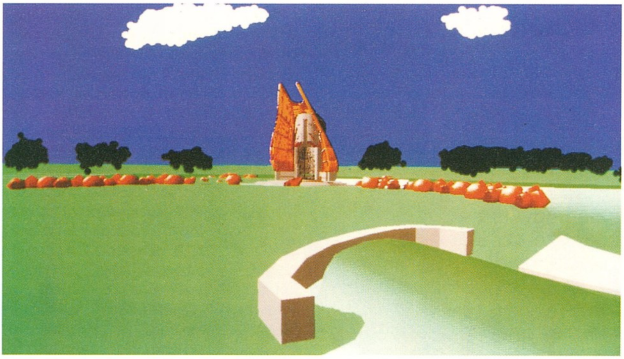
A modell építése, illetve folyamatos alakítása, finomítása másfél hétig zajlott a számítógép képernyőjén. Az első rázósabb feladatot a térben változó geometriájú főtartók „felszerkesztése" jelentette, ahol a ráfeszülő héjszerkezetet analitikusan nem is lehetett „megfogni". Az általános célú építészeti CAD rendszerek rendkívül jól szerkesztenek alapformájú tetőket (sátor-, kontytető, ...), de kilépve a „kocka-építész" szemléletből, a feladat jóformán megoldhatatlanná válik. A választott CAD rendszerrel (Point Line) a feladat ténylegesen megoldható. Bár az építtető két nézetben megfogalmazott elvárásait nehéz volt „lefordítani" a gép ábrázoló, geometriai nyelvére, végül azonban térben megfogva az egyes sarokpontokat, minden a helyére került, és kialakult a kápolna alapformája.

Izgalmas feladat volt a színharmónia kialakítása a 3 alapszín kikeverésével. Tudjuk, hogy valamennyi alapszín egy 256-os skálával rendelkezik. Ezekkel a kis színváltásokkal is sokat módosíthattunk a felhasznált szín árnyalatain. Így rengeteg pozícióból számtalan látványképet néztünk meg, mielőtt kimondtuk a végső szót.

Magas fokú CAD-kultúra

Ezután készítettük el azt a 30 másodperces videofilmet, amely a még meg sem valósult kápolnát mutatja be. Így például a szponzorok jobban el tudták képzelni, hogy mit is támogatnak valójában. Ez az a pillanat, amikor minden felhasználó számára megfogható, hogy mennyire pótolhatatlan a számítógép. Ugyanis, ha hagyományos módszerekkel készítik el ugyanezt a filmet, akkor az 1-2 hónapba is beletelik, és a számlán szereplő végösszeg közel tízszerese a számítógépes tervének. És még nem beszéltünk a számítógép azon „alaptulajdonságáról", hogy az így elkészült modell lesz az alapja a későbbi műszaki rajzoknak. Így készülhetett el — a szűkös határidő miatt — egyetlen hosszú nap alatt az összes engedélyezési terv is.

További két héten keresztül már csak a részleteket finomítottuk. Elkészült a fatartó, a kövek gyártási tervei, az épület kiviteli terve. Ezek a tervek a térbeli számítógépes modell segítségével pillanatok alatt „előálltak", és nem tértek el a korábbi tervektől. Hagyományos módszerekkel ezeket a térbeli szerkezeteket úgyszólván nem lehetett egzaktul definiálni, csak megközelítően: fa- és papírmodell segítségével. Bár a számítógépes modellben is vannak hibák,



de ezek nagysága legfeljebb 1 centiméter, szemben a hagyományos megközelítéssel, ahol akár 10 centiméternél nagyobb hiba is elképzelhető. A hazai CAD-kultúra fejlődésének jelentős lépésének éreztük — így nagyon örültünk —, amikor a statikus floppyn kérte az adatokat. Most joggal mondhatjuk, hogy léptünk egyet Európa felé.

Nem szeretnénk a felhasznált Point Line rendszert dicsérni, egyes tudáselemeit kiemelni. Úgy gondoljuk, elegendő elmondanunk, hogy az első pillanattól az utolsóig számítógépes platformon készültek a tervek. Ceruzát csak az egymás (az építész és a számítógépes tervezőmérnök) közötti kommunikációra használtunk.

Nem tudjuk, hogy számítógép nélkül ilyen jellegű feladatok hogyan lennének egzakt módon megrajzolhatók. Most már azonban tudjuk, hogy számítógép, és a megfelelően kiválasztott szoftver segítségével hogyan és milyen nehézségek árán kódolhatjuk az információkat. Ilyen és más hasonló feladatokon okulva megtanultuk, hogy a kritikus pont az építész agya és a szoftver közti interfész. Még csak álom, hogy az egyik irányban tökéletes legyen ez a kapcsolat, de az ideális a kétirányú kommunikáció lenne!

Kuczogi László

OrCad II-szoftvertámogatás I.

Öreg szoftver nem rossz szoftver

Bizonyára sokan csodálkoznak rajta, hogy miért foglalkozunk ezzel, a ma már klasszikusnak számító NYÁK-tervezőrendszerrel. Az ok kézenfekvő: még mindig a legnépszerűbb program a maga kategóriájában, szoftvertámogatása pedig hagy némi kívánnivalót maga után. Célunk, hogy röviden — a régi OrCad-esek mellett a programmal most ismerkedőket is szolgálva — azokat az utasításokat nézzük át, amelyek a legfontosabbak. Kétrészes ismertetőnk közreadásával úgy véljük, konkrét gyakorlati segítséget adhatunk a népszerű szoftver használatához.

Kezdve az alapoknál...

A Draft (rajzoló-) program alapfokú használata nem bonyolult. Kevés angoltudással és találékonysággal könnyen készíthetünk rajzokat. A program sok olyan lehetőséget is kínál, amelyek használata ugyan nem szükségszerű, de a rajzolást gyorsabbá teszi.

Ezeknek a funkcióknak a leírása előtt azonban nézzük meg a program konfigurációs képernyőjének érdekesebb utasításait! (A teljes leírás a lemezmellékleten megtalálható.) A konfigurációt elég csak egyszer beállítani, ezt a program a továbbiakban rögzíti. A még be nem állított konfiguráció esetén a program indításakor automatikusan a konfigurációs képernyő jelenik meg. Ha egy beállítást akarunk megváltoztatni, akkor a képernyőt elő tudjuk hívni a /C kapcsolóval. A kapcsolót természetesen a plotter-, a printer- és a többi segédprogram neve után is írhatjuk. Ilyenkor ezek előtt a programok előtt jelentkezik be a menü. A konfigurációs képernyő teszi lehetővé a gépünknek megfelelő meghajtók, a rajzoláshoz szükséges papírméretek, színek, a használandó alkatrészkönyvtárak beállítását.

Az LF utasításban rögzítjük azoknak az alkatrészkönyvtáraknak a nevét, amelyekkel majd dolgozunk. Ezekből a fájlokból azonban nem adhatunk be csak egy adott memóriaterületet elfoglaló mennyiséget. (Ajánlott például: CMOS.LIB, TTL.LIB, ANALOG.LIB, DEVICE.LIB beadása.) A CT utasítás kiadása után állíthatjuk be a rajzoláshoz használt színeket és tollméreteket. A TT utasítással pedig a rajz és a rajzelemek méreteit adjuk meg. Itt mondjuk meg (collban) a különböző rajzlaptípusokhoz (A, B, C, D, E) tartozó méreteket, a felirat-, a rajzlapkeret méreteit, a háló adatait.

A funkcióbillentyűket használhatjuk saját céljainkra is, például a program ennek hatására lefuttat egy utasítássorozatot (makrót). Ezeket mi is megírhatjuk a DRAFT programban (lásd később), s ezt egy makrófájlban rögzíthetjük, és az MF utasítás után adhatjuk meg. Az itt definiált makrót már nem kell a rajzolóprogramból külön READ utasítással beolvasni. A program indításakor ugyanis egy makró azonnal lefuthat. Erre szolgál az IM utasítás. Ezt akkor érdemes használni, ha a DRAFT program indítása után például a hálót azonnal láthatóvá akarjuk tenni. Az U utasítás hatására a program rögzíti a megváltoztatott konfigurációt. Az R és Q utasítások esetében is — amennyiben változtattunk a konfiguráción — megkérdezi a program, hogy rögzítse-e az új adatokat, majd elindítja, illetve kilép a programból.

Folytatva a kuriózumokkal...

A rajzolóprogram alapvető funkcióit ismertnek feltételezve csak a különlegesebb utasításokat fogjuk megnézni

(SET, MACRO, JUMP). Az egyik legfontosabb utasítás talán a SET, amelynek segítségével beállíthatjuk a program paramétereit.

Auto Panning. A kurzor a képernyő szélére érve az ablakot arrébb görgeti a rajzlapon.

Drag Busses. A BLOCK utasítás DRAG módjában a BUS vezetékeket is továbbköti.

Macro Prompts. A makrók futtatása közben mutatja a hívott utasításokat.

Orthogonal. A vezetékeket függőlegesen és vízszintesen lehet fektetni.

Show Pins. Az alkatrészek lábszámozása látható.

X,Y Display. Kurzorkoordináták megjelenítése a bal felső sarokban.

Grid parameters. Rácsparaméterek.

G. References. Rajzlap-koordináták megjelenítése a lap szélein.

Stay On G. A kurzor csak a háló pontjain lehet.

Visible G. Dots. Láthatóvá teszi a hálót.

Repeat Parameters. X,Y Rep.

Step. A REPEAT utasítás mennyivel mozdítja arrébb a rajzelemet.

Label Repeat Delta. Feliratok ismétlésekor a bennük szereplő számot mennyivel növelje a REPEAT utasítás.

Auto Increment Place. LABEL PLACE utasítás többszöri használatakor a fenti értékkel növeli a feliratban lévő számot.

Visible Lettering. A Zoom 2-es értékénél milyen rajzelemek legyenek láthatóak.

Makrózva

A rajzolóprogram egy másik érdekessége a makróhasználat. A makrókat CAPTURE üzemmódban definiálhatjuk a az alábbiak szerint:

Először adjuk be a hívóbillentyűt! [Az összes funkcióbillentyű magában, valamint az összes billentyű SHIFT-tel, Ctrl-lel(^) és ALT-tal(\) szerepelhet.]

A kívánt billentyű(k) kiválasztása és az ENTER lenyomása után adhatjuk be a programot úgy, mintha használnánk az utasításokat. Ha végeztünk, akkor az 'M' billentyűvel léphetünk ki a makróból. Ezután a beadott program a megfelelő billentyűkombinációra elindul. Előfordulhat, hogy egy utasítás közben akarjuk befejezni egy makró futását — például vezetéket akarunk lefektetni az egér középső gombjával (MMB). Ilyenkor az M betű helyett a 'CTRL-END' kombinációt kell használni.

Ha olyan programocskát írunk, amely a billentyűzetről kér paramétereket, akkor használjuk a 'CTRL-HOME', 'ENTER' kombinációt a paraméter helyén! (Például az F1 hatására lépjünk be a GET utasításba, a billentyűzetről válasszunk egy alkatészt, és ezt tegyük le!)

Amennyiben egy makrót a konfiguráció automatikusan hív, akkor azt mindenképpen ENTER-rel kell kezdeni! A már kész makrók közül DELETE módban törölhetünk. Az INITIALIZE utasítás pedig az összes makrót törli. A LIST utasítás hatására a gépben lévő makrók hívógombjainak listáját láthatjuk. A READ/WRITE utasítás olvassa/írja lemezre a makrókat. Az OrCad2 programcsomag jó tulajdonsága, hogy tartalmaz saját makróprogramokat is (a MACRO1.MAC és MACRO2.MAC lemezmellékletünkön megtalálható).

Ugrálva

Két további hasznos utasítással is célszerű megismerkednünk, ezek a JUMP és a TAG utasítások. Az utóbbival referenciapontokat helyezhetünk el a rajzlapon (A-tól H-ig). Ezekre a pontokra a JUMP megfelelő módjában a kurzor automatikusan elugrik. A JUMP utasítással megkereshetjük a beadott pozíciószámot (REFERENCE), illetve ugorhatunk adott értékeket függőlegesen (Y LOCATION), illetve vízszintesen (X LOCATION). HARDCOPY módban a rajzot kiprintelhetjük, illetve a rajzból grafikus fájlt készíthetünk. Ezt később DOS-ból a COPY utasítással akár ki is nyomtathatjuk. Hogy mit csinálunk, azt a DESTINATION címszó alatt dönthetjük el (LPT, FILE). FILE módban választhatunk, hogy felül akarjuk-e írni a beadott fájlt (REPLACED) vagy bővíteni akarjuk (APPENDED) azt egy rajzzal. A WIDTH OF PAPER címszó alatt a papír állását választhatjuk ki, míg a MAKE HARDCOPY utasítással elindíthatjuk a nyomtatást (rögzítést).

A szoftver a leírtakon kívül természetesen még sok mindent tud. Ezek közül az érdekesebbekkel (a segédprogramokkal) behatóbban a következő számunkban foglalkozunk.

Záruba Károly

CAMP — kempingszéken

Kellő beharangozó reklám nélkül kissé belterjesre sikeredett az idei CAMP. S a szervezés körüli gondokat jól jelzi a címadó sztori is: az egyik kiállító cég az eredetileg egy nappal hosszabbra tervezett kiállítási terminust adta meg partnereinek, s mivel már túl későn értesült az egynapos „rövidítésről", nem állt módjában mindenkit tájékoztatni. S hogy a vidékről pénteken felutazó partner ne maradjon hoppon, a cég a Kongresszusi Központ lépcsőjén vert tanyát — háta mögött a már bezárt kiállítással, „ölében" pedig a kiállított portékával.

A bensőséges hangulatú kiállításon talán a legnagyobb nyüzsgés az Autodesk standján volt. Valószínű, hogy a területében legnagyobb kiállítóhoz nemcsak az AutoCAD Release 12 premierje vonzotta az érdeklődőket, hanem az a kedvezményes akció is, amelynek keretében az AutoCAD Release 11 magyar verziójához 180 000 helyett csupán 99 000 forintért juthattak a felhasználók. Az AutoCAD-híveket több apró öröm is érte a CAMP-en, hiszen itt találkozhattak az AutoSketch Windowsos verziójával, valamint azzal a shell programmal, amellyel az angol AutoCAD Release 11 Windows alatt fut.

A CAMP '92 dobogóján helyet foglaló Prime Computer Vision teljes termékskáláját felvonultatta. A legtöbben a 3D-ben parametrikusan programozható Cadds5 és a relációs adatbáziskezelővel integrált Medusa iránt érdeklődtek.

A CAMP legnagyobb hazai kiállítója, a CADserver Kft. a teljes CAD/CAM palettát lefedő, magas színvonalú szoftvereit mutatta be. Az érdeklődést tovább növelte, hogy akciót hirdettek a PADS NYÁK-tervező rendszer kedvezményes vásárlására.

Az „első bálozós" kiállítók között igazán népszerűnek bizonyult a Nemetschek Iroda, amely most mutatta be szélesebb körben a nekünk oly kedves nevű — itt azonban csupa nagybetűs — NEMETSCHEK tervezőrendszert. Ez az építészeknek és statikusoknak készült szoftver meglehetősen drága ugyan, de tartalmaz minden (!) olyan eszközt, amire az építész/építő mérnöknek szüksége van munkája során.

Az idei CAMP jellemzője a „finomítások" térnyerése volt. Nem falrengető újdonságok, hanem a már létező rendszerek továbbfejlesztett verziói kerültek reflektorfénybe. Ezek még inkább felhasználóbaráttá és komplexszé teszik a már jól ismert rendszereket (ArchiCAD, ArchiTECH.PC).

A szakkiállítás jól tükrözte a magyar ipar profilváltozásait is. Az ipari háttér megváltozása jelentősen csökkentette a gépészeti szoftverek iránti keresletet, de továbbra is népszerűek az építészeti, elektronikai rendszerek. Sokakat érdekelt az animáció és látványtervezés is. Szembetűnő volt azonban a klasszikus CAD-alkalmazások mellett a térinformatika előrenyomulása. A kiállítók legalább egyharmada — elsősorban az önkormányzatoknak adresszálva — kínált ilyen szoftvereket.

A kiállítók előszeretettel preferálták a Silicon Graphics, a Sun, a Hewlett Packard és a DEC platformokat. Több forgalmazó véleménye szerint a munkaállomáson futó szoftverek „kiszorítják" a PC-s rendszereket. Mások szerint azonban még mindig a PC-s változat érdekli jobban az embereket. Az ár- és teljesítményharc tovább tart.

Sziebig Andrea

Igazoltan

10 MILLIÓSZOR



AZ IPARI SZABVÁNY NYÚZÓTESZTJE HÁROMMILLIÓ ÍRÁSI-OLVASÁSI MŰVELETET KÖVETEL, EHELYETT A KAO EZT TÍZMILLIÓSZOR TELJESÍTI. AZ IPARI SZABVÁNY 45%-OS CLIPPING-SZINTET ÍR ELŐ, EZZEL SZEMBEN MINDEN KAO LEMEZ LEGALÁBB A 70%-OT ELÉRI. TEHÁT, HA A LEMEZEK, STREAMERKAZETTÁK ÉS DAT-OK MINŐSÉGE ÉS MEGBÍZHATÓSÁGA ÖNNEK FONTOS, AKKOR CSUPÁN EGY MÁRKANEVET TARTHAT SZEM ELŐTT. KAO.

KAO®

MAKROTREND - 1143 Budapest, Hungária krt. 65 - 67. Tel: 183-4356 Fax: 163-7888

INFORMÁCIÓKÉRÉS: 41 ▲

A négyek bandája

Tudósítás az időgépből

Kaleidoszkópunk eddigi forgatása során — elvégre mindnyájunknak sok dolga van számokkal — különös előszeretettel vettük szemügyre a számnevek kifejezési formáit a különböző nyelvekben. Talán a legmeglepőbb számképzési rendszerrel a „Digitális Bábel"-nek szentelt, 1992. februári számunkban ismerkedhettünk meg. A mi fogalmaink szerinti kerek számokról kiderült, hogy az alapul vett nyelvben csak nagyon bonyolult módon lehet őket kifejezni, és megfordítva, kevésbé kerek számoknak egyszerű kifejezési forma felelt meg. Mélyebbre ásva kideríthetjük, hogy a magyarázatot különböző alapú számrendszerek kereszteződésében lelhetjük fel... Mostani feladatunkból kiderül, hogy számrendszerek kereszteződése másutt is előfordul, sőt nem is olyan ritka jelenség, mint gondolnánk.

Olvasóink közül jó néhányan kitűnően megoldották februári feladatunkat (emlékeztetőül mágneslemez mellékletünkön felidézzük a feladatot, és idézünk olvasóink megoldásaiból), magának a rendszernek a kialakulásával kapcsolatban azonban már jóval kevesebben merészkedtek hipotéziseket felállítani. Ami egyébként nem meglepő, hiszen tudósaink és művelődéstörténészeink sem könnyen igazodnak el a világ nyelvein belül a számnevek önmagában is bonyolult világában. Pedig a hipotézisek felállítása, a sejtések megfogalmazása szervesen hozzátartozik a tudományok fejlődéséhez, akár beigazolódnak később, akár éppen a megcáfolásuk igénye serkenti további erőfeszítésekre a tudóstársadalmat.

Miért éppen negyven?

Legfontosabb információnk, hogy milyen számokat tekint ez a nyelv „kerek", kitüntetett számoknak. A megoldásból látható, hogy először is a negyvenet. Önmagában ez a tény nyilvánvalóan kevés egy konzisztens hipotézis felállításához, de kiindulópontnak mindenképpen sokatmondó.

Próbáljuk kiterjeszteni ezt az információt más tudományágak, a néprajz, a nyelvészet, és nem utolsósorban a mitológia segítségével. A mitológiát mint hasznos segédtudományt nem szabad lebecsülnünk a múltra vonatkozó hiányos ismeretek lyukainak kitöltésében, hiszen gyakran ebből deríthetünk ki olyan kulturális kapcsolatokat, amelyeknek materiális nyomai, ha egyáltalán fennmaradtak, sokkal nehezebben bírhatók szóra.

A negyvenre vonatkozóan sok érdekes nyomot találhatunk több nép hagyományaiban és mitológiájában. Mezopotámiában a negyven az akkád Enkinek, a vizek istenének volt a szent száma. Ez különösen érdekes annak a fényében, hogy a Biblia szerint az özönvíz idején negyven nap és negyven éjjel zuhogott az eső. (Sőt a múltba sem kell mennünk: mindnyájan úgy tudjuk a népi meteorológiából szerzett ismereteink alapján, hogy negyvennapos esőzést jelent a Medárd-napi eső. Pedig ha az esős időszak beköszöntésének lehet is valóságos tapasztalati alapja, annak már aligha, hogy ez éppen 40 napig tartson.)

A negyven mint kitüntetett szám másutt is gyakran előfordul a Bibliában a „meghatározott sok" kifejezésére. 40 évig bolyongtak a zsidók a pusztában. 40 évig uralkodott Dávid, Saul és Salamon. Jézus 40 napig böjtölt és imádkozott, majd halála után 40 nappal szállt föl a mennyekbe. A példák számát folytatni is lehetne, véletlenről szó sem lehet.

Az arabban is megtalálható a népmesei motívumok között a negyven; hogy csak a nemzetközileg ismert példára hivatkozzunk, Ali babával kapcsolatban mindnyájan tudjuk, hány volt a sok rabló.

A nyelvészeti példák közül figyelmet érdemel a sok vitát kiváltott „szorok" szó az oroszban, amelynek az ukránon kívül más szláv nyelvekben nincs nyoma. Fogalmi körével kapcsolatban érdekes, hogy a százlábú neve az oroszban szorokonozska, vagyis „negyvenlábú". Megszámlálhatatlanul sok, tehát negyven.

Mindezek a példák azonban elszigetelt esetek, rendszerszerűség nem figyelhető meg rajtuk keresztül a számolás rendszere, és a negyvennek mint számfogalomnak a kifejezése között.

Te is fiam, négyszáz?

Ritka szép karriert futott be az iráni hazar szó, amely — a nyelvészek jól tudják — a magyar ezer szónak is a forrása. Eredeti jelentése ennek már a soknál is több: töméntelen, rengeteg. „Ezer" jelentésben megvan a Kaukázus tájékán az örményben és az oszétban. (Lehet, hogy mi is ez utóbbiaktól vettük át, mint az „asszony" szavunkat. Tudták a jászok, kiknél kell házát keresniük.) Érdekesebb azonban, hogy a Pamír hegység kies vidékein is felbukkan a hazar, azar, yekzar szó különböző nurisztáni és dard nyelvekben. E nyelveket beszélő népek általában húszas egységekben számolnak, így a száz nekik 5 húsz, a kétszáz 10 húsz, a háromszáz 15 húsz. Ahol a hazar szó különböző alakváltozatait használják számrendszerükben (így például a dard gawar és a nurisztáni ashkun nyelvben), ott ez rendszerint a húszas rendszerben értelmezett magasabb szintű egységet, vagyis a négyszázat jelenti. Az idegenektől eltanult szót tehát saját szokásaikhoz idomították, saját rendszerükbe illesztették bele.

Ha nem is a teljesség, de a számos nyelvben való elterjedtség szemlélteté-

sére megjegyezzük, hogy a mai napig megtalálhatók egy hasonló húszas számrendszer nyomai az ősi kultúrával rendelkező Délnyugat-Afrikában, elsősorban Ghánában és Nigériában, és főleg a szudáni nyelveket beszélő népesség között. És hogy a kép még tarkább legyen: húszas számrendszerben számolnak az eszkimók is — nem valószínű, hogy a közhiedelemnek eleget téve, mármint a lábuk ujjait is felhasználva a számoláshoz. De más nyelvekből is lehetne példákat idézni. Hogy egy kevésbé közismert példára hivatkozzunk: Abraham Lincoln még bizonyára minden angol származású honfitársa számára érthetően használta — egyik beszédében, alapszámként — a ma már elavultnak tekinthető score szót, mikor éveket jelölő számnevet akart körülírni: „Four score and seven years ago..."

Minél inkább vissza

Mindezen kutatásaink azonban nem vezetnek el eredeti problémánk megoldásához. Egyazon rendszerben a negyven és a négyszáz (az általunk megszokott értelemben) csak akkor fordulhat elő, ha közöttük a váltószám 10. Ugyanezt erősíti meg a négyezernek ugyanilyen rendszerszerű alkalmazása is. Figyeljük meg: ha alapegységként a februári feladatban megismert számrendszerben a 40 értékű számot vesszük, akkor egy egészen közönséges 10-es számrendszerhez jutunk. „Rendszeridegennek" azt az alrendszert kell tekintenünk, amely a negyvenig való számolást jelenti.

Ezzel az ötlettel pedig már el is jutottunk a talány megfejtéséhez: olyan rendszerek egybenövésére kell gondolnunk, amelyben vannak valamilyen kis egységek a kis mennyiségek megszámolásához, a nagyobb mennyiségeket pedig már egy nagyobb egységhez viszonyítva fejezik ki.

Hol láthatunk ilyen rendszereket? Mindenütt a világon, ahol pénzt használnak. Vannak valamilyen kis értékű váltópénzek a viszonylag kis értékek kifejezésére, és van ennek valamilyen egész számú többszörösével kifejezhető értékű pénzegység a valóságos számoláshoz. Egyáltalában nem kötelező, hogy a két pénzegység közötti váltószám ugyanaz legyen, mint az a számrendszer, amelyet az „igazi" pénzegységgel való számolásban használnak.

Ezek után már nem is kell sokáig kutatnunk, hogy megtaláljuk a hiányzó láncszemet. Ismeretes, hogy a legrégibb és legszélesebb körben elterjedt pénz Ázsiában, Afrikában és a hozzájuk közel lévő szigetvilágban ősidők óta a kauri kagyló (angolul: cowrie) volt.

A titok nyitja

Kauri ábrázolatok már a paleolitikus barlangrajzokon is feltűnnek. Egyiptomban a termékenység talizmánjának tartották a kaurit. A fáraósírokból milliószámra kerültek elő a finoman csiszolt kauri kagylók, amelyeket rendszerint füzérekké fűztek össze, hogy könnyebb legyen számolni velük. A szegények számára néhány szem kauri is komoly érték volt, a gazdagabbak azonban inkább csak a füzéreiket számolgatták. Kétségtelen, hogy a mindennapi életben a felfűzetlen kaurik töltötték be a váltópénz szerepét.

Több ezer éven keresztül messze földről, az Indiai-óceán melletti Maldíviából hordták karavánok a kauri kagylókat. Egyiptomból a Szaharán keresztül szállították Nyugat-Szudán területére, ahol a legnagyobb volt a becse. Később angol és holland kereskedők is bekapcsolódtak az üzletbe, és hajókkal vitték a kaurit Guinea kikötőibe, bár nem nagyon értették, hogy miért becsülik az afrikaiak többre, mint az aranypénzt. (Menyasszony megváltásához vagy adófizetésre el sem fogadtak egyebet.) A 20. században elindult ásatások elsősorban a joruba nép szellemi és kulturális központja közelében, Ife (Ilife), a szent város tájékán találtak rengeteg kauri kagylót.

Nem csoda ezek után, hogy éppen Afrikában maradt fenn legtovább a kauri pénz. Egyes kutatók még a 20. század negyvenes éveiben is találkoztak vele. A század elejéről származó leírások pedig azt igazolják, hogy ahol az ősi szokásokat, a Törvényt pontosan betartották (elsősorban a guineai tengerparton és Nigériában), ott éppen 40 kauriból készült a füzér.

A negyven kauri számolását rendszerint ötösével végezték. Később, amikor elkezdődött a pénzrontás, éppen öt kaurival rövidítették meg azokat, akik számára a füzér volt az egység. (Érdekes nyelvi adalék: Homérosznál fordul elő a pempadzo, pentadzo „ötösölni" ige 'számlálni' jelentésben.)

Nagyon valószínű, hogy a húszas számrendszer és a negyvenes füzérbe fűzés nem volt független egymástól. Hogy mást ne mondjunk, joruba nyelven ogodzsinak nevezik a füzért, ami tulajdonképpen „két húszast" jelent. A képhez hozzátartozik, hogy a yorubák négynapos hetekkel mérték az idő múlását, továbbá hogy a 4-nek és a 4 hatványai közül főleg a 16-nak és a 256-nak (egyes kutatók szerint a 4096-nak is) központi szerepe volt a joruba papok híres jóstudományában, akiktől Orunmila isten tanácsait kérték ki fontos döntéseknél.

Mindezek után lássuk az új feladatot. Egészítsük ki az alábbi egyenlőségeket, majd próbáljuk kifejteni gondolatmenetünket:

1. femxfem=femotyve
2. fireofirsinstyve+seks=halfemsinstyve
3. seksotresinstyve+niden=femofirsinstyve
4. femden+femotresinstyv=firsinstyve
5. treden+=niotyve
6. seksxni=
7. niotresinstyve+fireotyve=

Melyik nyelvről lehet szó?

Versenyünk újabb fordulója 1993. áprilisi feladványunkkal ér véget. Előző Kaleidoszkóp versenyünkhöz hasonlóan minden szám után létraversenyszerűen összesítjük a szerzett pontokat. Előzetes tervünk szerint 1993. júliusában lesz az eredményhirdetés. A „létra" felső fokaira kerülteket díjazzuk, de hogy hány jutalmazott lesz, az attól is függ, milyen aktivitással vesznek részt olvasóink a versenyben, továbbá hogy milyen minőségű megfejtéseket küldenek be. (Ha még emlékeznek rá: múltkori versenyünkben olyan kiváló megoldásokat kaptunk, hogy előzetes ígéretünket túlteljesítve jóval több jutalmat osztottunk ki.)

Ismét elmondjuk: a feladatok tematikájához kapcsolódva módot szeretnénk nyújtani az olvasók egyéni aktivitásának kifejtésére is. Rövid programokat, programozási fogásokat szívesen közlünk mágneslemez mellékletünkön, bőséges kommentárokkal. Felhívjuk a figyelmet Programozástechnika rovatunkra, ahol már elkezdtük a Snobol nyelv ismertetését. Ez a nyelv kiváló eszközt jelenthet nyelvi feladatok viszonylag könnyű megoldására (ráadásul sok más nem aritmetikai probléma elegáns megoldására is).

Természetesen sok feladat más nyelvekben is ügyesen megoldható — kérjük olvasóinkat, hogy hívják föl erre a mi figyelmünket is, és rajtunk keresztül többi olvasóinkkal is osszák meg ismereteiket.

Mostani feladatunk megoldásának beküldési határideje: **1992. november 10.** A megoldásokat a szerkesztőség címére kérjük eljuttatni: Alaplap szerkesztősége, Kaleidoszkóp, 1441 Budapest, Pf. 74.

Vargha Dénes

FAN computer

PROFESSZIONÁLIS SZÁMÍTÓGÉPEK

4 ÉV GARANCIÁVAL

MS DOS 5.0-val és WINDOWS 3.1-gyel is!

VELÜNK VÁLTSON SEBESSÉGET!

- 286/25–33 MHz számítógépek (bővítés: 32 MB-ig!),
- 486/50 MHz számítógépek,
- WINDOWS-gyorsító kártyák,
- CACHE – IDE vezérlőkártyák (százszoros hozzáférési sebesség).

SZÍNES ÉS MONOKRÓM,
ASZTALI ÉS KÉZI SZKENNEREK, EGEREK,
DIGITALIZÁLÓ TÁBLÁK

FAN Electronics Ltd

Tajvani–Magyar Vegyes vállalat
1118 Budapest, Késmárki u. 6.
(volt Friss István u.) Telefon/Fax: 185-0813

INFORMÁCIÓKÉRÉS: 23 ▼

DATENTECHNIK

Kereskedelmi Képviselet
Budapest I., Naphegy tér 8. 1016
Tel./Fax: 175-0182

Programozható, intelligens modemek kapcsolt telefonvonalra, automata hívóművel, hívószámtárolóval, hibavédelemmel, adatkompresszióval, széles sebességhatárok között: 300–19 200 bps. Bérelt vonali modemek 2–4 huzalos áramkörökre, széles sebességtartományban.

INFORMÁCIÓKÉRÉS: 15 ▼

DAXON

elektronikus

KULCS A SZÁMÍTÓGPÉHEZ

hozzáférés-védelem

Próbálja ki!

DAXON Elektronikai Kft.
1114 Budapest, Eszék u. 12.
Telefon: 161-3366 • Fax: 161-3339

INFORMÁCIÓKÉRÉS: 16 ▼

WACH és Fia Kft.
1093 Budapest IX., Bakáts u. 2/C
Tel./Fax: 137-2344, Tx.: 22-3756 wach h

NE DOBJA EL BESZÁRADT, KIÍRT, KIÜRÜLT FESTÉKKAZETTÁIT!!!

Cégünk eredeti amerikai „MACINKER™" technológiával, eredeti amerikai gépekkel és festékekkel vállalja valamennyi Magyarországon forgalmazott printer- és írógépkazetta felújítását, javítását, újrafestését STANDARD és OCR kivitelben, garanciával. A speciális technológiából adódóan saját csomagolásban 3 év raktározási garanciát biztosítunk. Az általunk felújított kazetták nem szennyezik a nyomtatófejet. Megrendelhető még CARBON kazetták, valamint HP LJET II, IIP, III, IIIP, SHARP, CANON, NEC lasercartridge-ok újratöltése is.

Nyitvatartás: 10.00–22.00-ig.

INFORMÁCIÓKÉRÉS: 56 ▼

ELENDER COMPUTER

ELENDER

Műszaki Kereskedelmi és Szolgáltató Kft.
1134 Bp. Csángó u. 13. Tel./Fax.: 129-9080
4029 Debrecen, Csapó u. 100. Tel./Fax.:(52) 13-795
6725 Szeged, Katona J. u. 9. Tel./Fax.:(62) 30-975

RENDKÍVÜLI SZÁMÍTÓGÉP AKCIÓ !

286/16 MHz-es számítógép ______ 46.900.-
1 MB RAM, 1,2 MB floppy, 40 MB Winchester, 14" mono monitor

386SX/33 MHz-es számítógép ______ 66.900.-
2 MB RAM, 1,2 MB floppy, 80 MB Winchester, 14" SVGA mono monitor

386/40 MHz, 64KB Cache számítógép ___ 98.900.-
4 MB RAM, 1,2 MB floppy, 120 MB Winchester, 14" SVGA color monitor

JETBOOK 386SX Notebook 114.900.-
386SX-20, 2 MB RAM, 40 MB Winchester, VGA LCD

JETBOOK 386DX Notebook 209.000.-
386DX-33, 32 KB Cache, 4MB RAM, 120 MB Winchester, VGA LCD

SANYO 386SX Notebook 119.900.-
386SX-20, 2 MB RAM, 60 MB Winchester, VGA LCD

Az árak ÁFA nélkül értendők, kp. fizetés mellett, 12 hónap cseregaranciával.

Ha számítógépet vásárol és ezt a szelvényt bemutatja, ajándékba egy egeret adunk Önnek.

ELENDER

INFORMÁCIÓKÉRÉS: 20 ▼

A C nyelv Unix felülete V.

A kényelem és a forma kedvéért

A Unix világa — mint más világok is — bizonyos ciklikus jelleget mutat: ahogy haladunk a közepe/sűrűje felé, vissza-visszatérnek a jelenségek/témák. Most megint a printelésről és az olvasásról tudunk meg dolgokat, de ezeket eddig nem érint(h)ettük.

A printf, fprintf, sprintf utasításokat módosíthatják (flag) karakterek. A % jel után el lehet helyezni formátummódosító (flag) karaktereket. A — karakter hatására balra lesz igazítva a mezőn belül az eredmény. A + karakter hatására az előjel mindig kiíródik (pozitív szám elé is). A szóköz hatására az előjeles mennyiségek elé vagy a negatív előjel, vagy egy szóköz íródik. A # karakter hatására az x (hexadecimális) formátumú számok elé 0x, az o (oktális) számok elé 0 íródik, ha az érték nem 0.

Példa:

```
x = 444;
printf("%8d\n",x);
printf("%-8d\n",x);
printf("%+8d\n",x);
printf("%-+8d\n",x);
printf("%d\n",x);
printf("% d\n",x);
printf("%#x %#o\n",x,x);
```

Kimenet:

```
     444
444
    +444
+444
444
 444
0x1bc 0674
```

Ha a mezőszélességet vagy a pontosságot megadó szám helyén egy * karakter áll, akkor az illető paraméter értékét az argumentumlistában soron következő (integer) elem fogja meghatározni:

```
f = 1.238; wid = 8; prec = 2;
printf("%*.*f\ñ",wid,prec,f);
```

Kimenet:

```
    1.24
```

A gets utasítással a standard bemenetről lehet egy sort olvasni:

```
char *gets(s)
char *s;
```

A sor végén lévő soremelés karaktert gets 0-ra cseréli. A függvény a beolvasott stringre mutató pointert ad vissza, hiba vagy EOF esetén pedig 0-t.

Az fgets utasítással egy fájlpointerrel megnyitott fájlból lehet olvasni:

```
char *fgets(s,n,fp)
char *s;
int n;
FILE *fp;
```

Az fgets mindaddig olvas, míg egy soremelés karaktert el nem ér, vagy (n—1) darab karakter be nem érkezett, ezután egy lezáró 0-t hozzáfűz a stringhez. (A soremelés karakter megmarad!) A fájl végének elérésekor a visszatérő érték 0. Ha a fájl nem létezik, vagy a hozzáférési jogok nem teszik lehetővé az olvasását, akkor a visszatérő érték szintén 0. Példaként vizsgáljuk meg a következő programrészletet, amellyel egy fájl tartalmát át lehet másolni egy másikba:

```
fp1 = fopen("DATA1","r");
fp2 = fopen("DATA2","w");
while(fgets(buff,1000,fp1)){
     fprintf(fp2,"%s",buff);
}
```

A puts utasítás a paraméterként kapott stringet a standard outputra másolja, továbbá hozzáfűz egy soremelés karaktert:

```
int puts(s)
char *s;
```

Az fputs utasítás egy fájlba másolja a paraméterként kapott stringet:

```
int fputs(s,fp)
char *s;
FILE *fp;
```

Egyik függvény sem írja ki a stringet lezáró 0 értékű bájtot. Az fputs utasítás ekvivalens a következő fprintf hívással:

```
fprintf(fp,"%s",s);
```

A getc egy makró (a /usr/include/stdio.h-ban van definiálva), tehát függvénynek nem adható át paraméterként. Segítségével egy karaktert lehet olvasni egy fájlból:

```
int getc(fp)
FILE *fp;
```

Az olvasott karaktert getc a visszatérő értékében adja át. Figyelem! getc nem char, hanem int típusú! Erre azért van szükség, hogy tetszőleges bájtot vissza tudjon adni, és jelezni tudja a fájl végét is. Az EOF érték —1 (l. /usr/include/stdio.h). Amikor sikeres az olvasás, a visszatérő érték legalsó bájtja az olvasott bájt, a többi pedig 0. Fájlvég érzékelése vagy hiba esetén getc -1-et ad vissza, tehát egy olyan 4 bájtos szót, amelyben minden bit 1-es. Példaként tekintsünk egy programrészletet, amellyel egy fájlt átmásolunk a standard kimenetre:

```
fp = fopen("DATA","r");
while((i = getc(fp)) != EOF){
     printf("%c",i);
}
```

Ha getc visszatérő értékét int helyett egy char típusú változóba töltenénk be, akkor a beolvasás befejeződne (tévedésből) a fájlban talált 255 értékű bájtnál. Ha fp nem érvényes fájlpointer, akkor getc —1-et ad vissza.

Az fgetc(fp) parancs ugyanazt csinálja, mint getc, de ez nem makró, hanem függvény, tehát átadható paraméterként. A getchar makró definíciója:

```
#define getchar()
getc(stdin)
```

tehát ez a standard bemenet olvasására szolgál.

A putc makró a paraméterként kapott karaktert kiírja fájlra:

```
int putc(c,fp);
int c;
FILE *fp;
```

A visszatérő érték a kiírt karaktert tartalmazza, vagy –1, ha nem sikerült az írás.

Az alábbi programrészlet karakterenként kiír egy stringet egy fájlra úgy, hogy az 'a' betűket 'A' betűre cseréli:

```
fp = fopen("DATA","w");
strcpy(buff,"abbacc");
for(i = 0,c = 'A'; buff[i];
i++){
     if(buff[i] != 'a')
putc(buff[i],fp);
     else   putc(c,fp);
}
```

A putchar makróval a standard kimenetre lehet írni egy karaktert. Az fputc függvény ugyanazt csinálja, mint a putc makró. Ha puffereletlenül akarunk a standard bemenetről olvasni, akkor természetesen a read utasítást kell használni:

```
read(0,buff,n);
```

Ha a standard kimenetre akarunk puffereletlenül írni:

```
write(1,buff,n);
```

Nemes Mihály

OOP — a Borland C++ 2.0 „színpadán"

„Gyakorlati" objektumok

Az elmúlt hónapban áttekintettük az objektumorientált programozás C++ szerinti megvalósításának fontosabb tulajdonságait.
Mostantól kezdve a gyakorlati kérdésekkel foglalkozunk. Ha bárkiben olyan technikai, szintaktikai, szemantikai stb. kérdés merül fel, amely közérdeklődésre tarthat számot, a szerző kéri azt a szerkesztőségbe.
A sorozat folyamán — vagy akár utólag — megpróbál minden felmerülő problémára választ adni.

A lemezmellékleten OOP02*.* név alatt vannak elrejtve az e részhez tartozó példaprogramok. Mielőtt a kedves olvasó belemélyedne a cikk tanulmányozásába, ezt célszerű kinyomtatni, vagy (ha nincs kéznél nyomtató) „bemutatni" valamelyik szövegszerkesztő programnak.

Az OOP-vel való ismerkedést egy klasszikusnak tekinthető példán keresztül fogjuk elkezdeni. Hozzunk létre egy struktúrát olyan értékek tárolására, amelyeknek mindig egy megadott tartományba kell esniük. Ilyen érték például az időpontnak az óra, perc és másodperc része, a mértani szögek értékei stb. Hozzuk létre a következő függvényeket a struktúra változóinak kezelésére.

1. void setlimit(int low, up) a határértékek beállítása;
2. int setnum(int num) egy érték igazítása és tárolása;
3. int setall(int num, int low, int up) az előző kettő egyszerre;
4. int addto(int num) tárolt érték megváltoztatása.

A feladat ANSI C megoldását az OOP2.C fájl tartalmazza. Ehhez semmi különösebb magyarázat nem tartozik.

Az első nekirugaszkodásra született C++ átiratot az OOP2_1.CPP fájlban rejtettük el. Nem csináltunk mást, mint hogy a struktúra adatain manipuláló függvényeket a struktúra törzsében definiáltuk. Ennek azonban több figyelemre méltó következménye lett.

Az ily módon definiált függvényeket függvénytagnak, vagy angolosabban member functionnek nevezzük.

A függvénytagok paraméterlistájáról hiányzik az a struktúrapointer, amely megmutatná, hogy az akciót melyik objektumon kell elvégezni. Ezt a mutatót a fordító automatikusan befűzi a paraméterlista elejére 'this' néven, explicit módon definiálni nem is szabad.

Továbbolvasva a példaprogramot, egy másik fontos különbséget fedezhetünk fel a funkciótagok kifejtésekor. Itt a :: (hatókör) operátor segítségével jelölni kell, hogy a kifejtésre kerülő funkció melyik struktúrához tartozik. Erre azért van szükség, mert a programban lehetnek azonos nevű függvények, amelyeket csak az különböztet meg egymástól, hogy más-más struktúrában — vagy éppen struktúrán kívül — lettek definiálva. Ha egy programban definiáljuk az X, az Y és Z azonosítójú struktúrákat, továbbá mindhárom struktúrában definiáljuk az F( ) függvényt, akkor az implementációkor az X::F( ), Y::F( ) és Z::F( ) fejléccel különböztetjük meg őket egymástól. Ugyanakkor a struktúrákon kívül definiált F( ) függvény kifejtését az ::F( ) fejléccel kell bevezetnünk. A :: (hatókör) operátor használata a funkciótagok és a valamelyik funkciótaggal megegyező nevű függvények kifejtésekor a fejlécben kötelező!

Az olvasással a funkciók törzséhez érve térjünk vissza a this nevű pointerhez. Tehát minden függvénymező törzsében (külön definíció nélkül is) hozzáférhető ez a struktúrapointer, amely arra a struktúrára mutat, amely a funkciót aktivizálta. Ilyen aktivizáló objektum pedig mindig van. Ahogy a példaprogramban is látható, ennek az elhagyása fordítási hibát eredményez. Emiatt a this sohasem lehet NULL pointer, hanem mindig egy létező objektumpéldányt címez meg. A józan ész azt diktálná, hogy a függvény törzsében a tagokhoz való hozzáféréskor használjuk ezt a mutatót, ahogy az ANSI C változatban az explicit módon átvett mutatót használtuk. Erre azonban nincs szükség. A fordítóprogram minden olyan hivatkozás elé „odaképzeli" a this-> prefixumot, amely az eredeti struktúra mezőire vonatkozik. Ezenkívül még arra is van figyelme, hogy a meghívott funkciótagoknak továbbadja ezt a mutatót. Ez utóbbi szolgáltatást a setall( ) funkció törzsében vesszük igénybe a másik két funkciómező aktiválásakor. A this explicit továbbadására akkor lehet például szükség, ha egy nem struktúratag-funkciót hívunk. Ebben az esetben a pointer továbbadása nem automatikus.

A main törzsében láthatjuk, hogyan lehet a struktúratag-funkciókat aktivizálni. Ha a konkrét objektumpéldányra vonatkozó prefixumot elhagyjuk, akkor a fordítóprogram egy struktúrákon kívül definiált függvényt keres, és ha talál, akkor oda adja a vezérlést. Ha ilyet nem talál, akkor egy goromba üzenet kiírása után dolgavégezetlenül kilép. Ez a mechanizmus biztosítja, hogy a this pointer mindig egy megfelelő típusú objektumra mutasson.

A következő lépésben vizsgált átiratot az OOP2_2.CPP fájl tartalmazza. Az eltérés az előzőekhez képest: a struktúra definíciójában használjuk a hozzáférésmódosító kulcsszavakat. A private: jelző után felsorolt tagokhoz kizárólag a struktúrán belül definiált függvények férhetnek hozzá. A public: utáni tagok hozzáférése teljesen szabad, megegyezik az ANSI C struktúramezők hozzáférési szintjével. A struct kulcsszóval bevezetett struktúrák mezői alapértelmezésben public: szintűek. Ez tette lehetővé az előző verzióban a hozzáférés-módosítók elhagyását.

Új kulcsszó a C++ nyelvben a class. Funkciója teljesen megegyezik a már ismert, és itt is korlátozás nélkül használható struct kulcsszóéval, azzal a nüansznyi különbséggel, hogy mezőinek alapértelmezett hozzáférési szintje private:.

A három adatmező privátkénti definíciója kizárja, hogy egy intrikus és illetéktelen utasítás „belerondítson" az ntrvlm típusú objektumainkba. Enélkül

ugyanis semmi garancia sincs rá, hogy valaki egy intervallumon kívüli számot írjon a number változóba, vagy akármilyen más gaztettet kövessen el a struktúra függvényeinek megkerülésével. Ez az adatrejtés azonban még azt is lehetetlenné teszi, hogy az osztályon kívülről akárki csak kiolvassa a változók tartalmát. Ezért szükségesnek látszott definiálni három új függvénymezőt, amelyek nem vesznek át paramétert, és visszatérési értékül a tárolt adatmezők tartalmát szolgáltatják.

A következő átirat az OOP2_3.CPP fájlban található. Itt újdonság a fentiekben már említett class kulcsszó használata. A class is struktúrát jelöl, akár a struct, de az előbbinél a mezők alapértelmezésben private: szintűek, míg az utóbbi alapértelmezése a public:.

A másik továbblépés az, hogy egyes függvények törzsét már a struktúra (már az osztály) definíciójában ki lehet fejteni. Az ilyenformán implementált függvényeket az inline jelzővel illetjük. Ez a fordító számára annyit jelent, hogy azt a kódot, amely ennek a függvénynek a fordításából származik, minden egyes hivatkozás helyére be kell másolni (ellentétben a nem inline függvényekkel, amelyek külön rutinba kerülnek, és a hivatkozásaik helyére rutinhívást illeszt a fordító). Ebből adódóan az inline függvények nagyon hasonlítanak a makrókra. Ezek is, azok is minden egyes hivatkozáskor teljes terjedelmükben bemásolódnak a kódba. Ha 213 esetben használjuk őket, akkor 213 példányban vannak jelen a programban.

A nagy hasonlóság ellenére az inline függvények mégiscsak függvények. Feldolgozásukat a fordító végzi el, és a fentebb említetten kívül nem tesz különbséget inline és nem inline között. Annyira igaz ez, hogy az inline definíció nem előírás a fordítóprogram számára, csak javaslat. Ha ő úgy látja jónak, akkor az inline definíció ellenére normál függvényként kezeli őket. Ez nagyon hasonlít a regiszterváltozók kezeléséhez. Mindkét esetben a programozónak csak „javaslattételi" joga van, a végleges (és megfellebbezhetetlen) döntést a fordítóprogram hozza meg.

Inline függvények létrehozásának csak egyik módja a már említett lehetőség, miszerint az osztály vagy struktúradefiníció belsejében fejtjük ki a funkció törzsét is. Ez az úgynevezett implicit inline definíció. Az explicit definíció kulcsszava mi más lehetne, mint 'in line'. E kulcsszó használatával bármelyik függvényre vonatkozólag inline feldolgozást kérhetünk.

A példaprogramban az implicit inline definícióra a get..( ) és a setall( ) függvények, az explicitre pedig az addto( ) függvény kapcsán mutatok példát. A kétfajta inline definíció között funkcionálisan semmilyen különbség nincsen, hatásuk teljesen azonos.

Fridl György

Hogy mondják Snobolul?

Csodatömb és társai

Előző számunkban már futó betekintést nyerhettünk a Snobol boszorkánykonyhájába.
Aki vette magának a fáradságot, és a mágneslemezen lévő két rövid programot is végigelemezte, az már szerzett némi fogalmat arról, hogy a frappáns fordulatokban gazdag programok megírása nem is kíván olyan óriási nagy nekiveselkedést.

Egyelőre megmaradunk eddigi témakörünknél: a nyelvstatisztikánál. Nem mintha egyéb alkalmazásai nem lennének a Snobol nyelvnek, de jobb, ha előbb egy meghatározott környezetben otthonosabban mozgunk, így aztán a sikerélményektől magabiztosabban próbálkozhatunk másféle problémák megoldásával.

Sokféle érdekes tanulsággal jár különböző jellegű szövegek betűgyakorisági statisztikájának az összehasonlítása — dehát ki veszi magának a fáradságot, hogy felkutassa a könyvtárakban az ilyen vizsgálatokról írt beszámolókat? Másképpen állna a dolog, ha a saját gépünkön magunk készíthetnénk efféle megfigyeléseket a magunk kiválasztotta szövegeken, vagy saját irományainkon. Mert kézzel strigulázni aztán igazán snassz dolog volna a számítógépek korában... Viszont ennek a kedvéért bonyolult programírásra rászánni magunkat? Na ne.

Legalábbis a hagyományos programnyelveken ne, ahol ezer gödröt kell kerülgetni még a legegyszerűbb programok elkészítésének kedvéért is.

Nos, éppen az ilyen egyszerűbb problémák számítógépes megoldásában érdemes elsősorban a Snobol segítségéhez folyamodnunk. Ha már kezdetben is néhány óra alatt eljuthatunk a megoldáshoz, később pedig negyedórák alatt, akkor bizony érdemes lesz gyakrabban leülni a számítógép mellé. Olyan esetekben is, amikor eddig legfeljebb egy futó gondolat cikázott át az agyunkon, hogy ezt milyen jól meg lehetne gépen oldani — ha éppen most nem sajnálnánk tőle a fáradságot.

Maradjunk meg az egyszerű, egyjegyű betűk területén, a többjegyűek kiválasztásával járó gondokat bízzuk a szakemberekre. (Ők sem tudnak velük mindig precízen elbánni.) Persze ha valakinek van kedve, néhány hónap múlva már könnyedén ír Snobolban olyan programot, amelyre rá lehet fogni, hogy hangstatisztikai vizsgálatokat végez.

Másban viszont nem alkuszunk. Azt, hogy a vizsgálat terjedjen ki az ékezetes magyar betűkre is, már most alapkövetelménynek tekintjük.

Mint minden hasonló problémánál, először az adatbevitel nehézségeivel kell megküzdenünk. Erre láttunk már példákat eddigi Snobol-tanulmányaink során, de azért nem árt egyszer alaposabban végiggondolni sorjában azoknak az utasításoknak a hatását, amelyekkel a nyers karaktersorozatból kiszabjuk a műtőasztalra fektethető, operációra kész „egyedeket".

A beolvasó ciklus megszervezése

1. A régi nóta

Múltkori feladatunkban úgy szerveztük meg a bemenetet, hogy egy „sor" nevezetű változóba vettük át az inputból rekordonként az információkat:

```
olvas sor = input                :f(kesz)
```

(Mire is kell vigyáznunk az utasítássor leírásánál? Ugye emlékszünk: a címkéknek mindig az első pozícióban kell kezdődniük.)

Gondoljuk meg, mi történik itt ennek az utasításnak a hatására. Érdekes: elég az értékadó utasítás jobb oldalán egyszerűen megemlíteni az „input" szót, ami itt most egy input elnevezésű szabványos változónak tekinthető.

Ez a forrása az inputnak, innen meríthetünk újabb és újabb adagokat valamilyen magunk választotta nevű változóba. Ha ezt a merítést ciklusban alkalmazzuk, a változóba átmentett rekord helyére a legközelebbi alkalomra már ott is a következő rekord, vagyis a következő sor. Mikor pedig eljutunk a fájl végére (az EOF jelhez), akkor az utasítás értelmében a „hibaágon" továbbadódik a vezérlés a „kesz" címkéjű utasításnak.

A rekordon belül a feldolgozás szavanként folytatódott. Ehhez szavakra kellett darabolnunk a rekordot; ezt egy belső ciklusban végeztük el a „kovszo" címkéjű utasítással:

```
kovszo sor szokep =              :f(olvas)
```

A címke utáni első karaktersorozatnak itt teljesen más a funkciója, mint az utána következőnek. A „sor" elnevezésű változó, láttuk, arra a rekordra utal, amelyet az imént vettünk át az inputból. Ez lesz a tárgya, az áldozati báránya a tüstént kezdődő műtétnek. (Angolban szubjektumnak nevezik, ami eléggé félrevezető elnevezés. Sajnos.) A következő változó nem egy egyszerű füzérre, hanem egy jóval bonyolultabb lelkivilágú valamire hivatkozik. Ennek az adattípusa is más: minta, angolul pattern. Az utasítássorban az áldozat után álló szó a neve annak a mintának, amelyet alkalmazni kell a feláldozandó karaktersorozatra.

Számítástechnikailag ezzel az utasítássorral történik meg a második ciklus beleágyazása az elsőbe. Sikeres esetben ugyanis valamilyen másodlagos objektumokat állítunk elő — mintegy mellékhatásként — a „szokep" minta segítségével, ezeket tesszük el ciklusban egymás után a „szo" elnevezésű változóba. „Hiba" esetén viszont a másodlagos ciklusból visszaadódik a vezérlés az elsődlegesnek, és sor kerül a következő rekord átvételére.

A minta kidolgozásának technikai részleteit most nem ismételjük meg, alkalmazásának sokrétű funkcionális szerepét viszont nem hagyhatjuk említés nélkül. A mintaillesztés eredményeképpen különböző dolgok történhetnek a minta szerkezetétől függően. Jelenlegi mintánk nyomán a következők történnek:

(1) előreszaladunk a „láthatatlan kurzorral" a legközelebbi szó elejére,

(2) megkeressük a szó végét,

(3) a kurzormozgással kiválasztott szót beletesszük a „szo" elnevezésű változóba.

Magával a mintaillesztéssel azonban még nem ér véget a műtét. Figyeljünk fel az egyenlőségjelre az utasításban, annál is inkább, mert a Snobol programok írásánál ennek elhagyása a leggyakrabban előforduló hiba. Mint látjuk, itt az egyenlőségjel jobb oldalán nem áll semmi, vagy — mondjuk úgy — üres szó áll az értékadó utasítás jobb oldalán. Ennek hatására a minta segítségével az imént kiválasztott karaktersorozat törlődik a „sor" elnevezésű változóból.

Ha ezt elfelejtjük, könnyen kerülhetünk végtelen ciklusba, mert újra meg újra végrehajtjuk (lényegében eredménytelenül) ugyanazt a tevékenységet. Szabályos esetben az „áldozati változó" tartalma már más lesz, mikor legközelebb átadódik a vezérlés ennek az utasítássornak.

Amíg el nem fogynak a szavak, addig mindig a következő utasításnak adódik át a vezérlés, amint azonban elfogynak, már más folytatást kér a „kovszo" címkéjű utasításnak a kettősponttal bevezetett GOTO része. Most az a ki nem mondott rendelkezés, hogy csak SUCCESS (s) esetén menjen át simán a vezérlés a következő utasításra. FAILURE (f) esetére ott áll fehéren-feketén az előírás: el kell ugrani az „olvas" címkéjű utasításra, hogy beolvassuk a következő rekordot. Hogy a rekordok elfogyása után mi történjék, arról már az elsődleges ciklus GOTO részének kell gondoskodnia.

2. Az új módi

Mostani feladatunk betűegységek kiválasztását kívánja meg. Megtehetnénk, hogy az eddigi két ciklust változatlanul hagyjuk, és beépítünk a másodikba egy harmadikat is a szavak betűkre bontására. Ha azonban a szavakra mint feldolgozási egységekre a későbbiekben nincs szükségünk, megtehetjük, hogy a második ciklus helyett rögtön olyan ciklust építünk be, amely egyenesen karakterekre bontja a sorokat. Igaz, ezek között nem betű karakterek is vannak, ezeket tehát valamilyen módon ki kell majd szűrnünk.

A változatosság kedvéért járjuk most ezt az utat. Nyilvánvaló, hogy a „szokep"-nek nevezett minta helyett kell egy másféle mintát alkalmaznunk. Az új mintát „karkep"-nak nevezhetjük, arra utalva, hogy alkalmazásával karaktert fogunk lehasítani.

```
karkep = len(1) . kar
```

A LEN(n) függvény mintákban alkalmazható: az a szerepe, hogy n karakterrel előremozgassa a „láthatatlan kurzort". Az így végigsepert karaktereket sikeres esetben a pontoperátor elteszi a „kar" elnevezésű változóba. (A pontoperátor előtt-után szóköz!)

A programba úgy kell beépítenünk a mintát, mint az előbb (nem feledkezve el a törlést biztosító egyenlőségjelről):

```
egykar sor karkep =              :f(olvas)
```

Az utasítás GOTO része gondoskodik róla kimondatlanul, hogy sikeres szeletelés esetén menjen tovább a vezérlés a következő utasítássorra. Ha azonban a sor végére érkezünk, ez az utasítássor „hibát" észlel, és visszapasszolja a vezérlést az „olvas" címkéjű utasításnak.

Mivel azonban itt nagyon egyszerű mintát kell alkalmaznunk, közvetlenül beleírhatjuk magát a mintát is az „egykar" címkéjű utasításba a neve helyett:

```
egykar sor len(1).kar =          :f(olvas)
```

Nem kell a minta elemeit zárójellel összefognunk — megállapodás szerint tudjuk, hol kezdődik, és meddig tart a „műtendő" jelsorozat, vagy annak a megnevezése, és hol kezdődik a minta, vagy annak megnevezése. Ami ettől kezdve van, az az egyenlőségjelig (vagy a kettőspontig, vagy ha ez sincs, a sor végéig) a mintához tartozik.

Előfordulhat persze, hogy a műtendő már a műtét előtt darabokból áll — ilyen esetben azt kell zárójelbe tennünk.

Gépesített Hamupipőke

Szegény Hamupipőkénknek még külön kell válogatni a betűket a nem betűktől. A Snobol a maga részéről két mintaépítő függvénnyel siet munkájának megkönnyítésére.

1. Az ANY(str) függvény

Argumentuma vagy eleve valamilyen füzér, vagy olyasmi, aminek kiértékeléséből füzért kapunk. Tevékenysége a SPAN(str) függvényhez hasonlítható. Kettőjük között az a különbség, hogy

a SPAN maximális hosszúságú „homogén" füzér kiválasztására szolgál (ha homogén füzéren azt értjük, hogy csak olyan karakterek tartozhatnak bele, amelyeket a SPAN operandusa kijelöl), az ANY pedig egyetlen karakter hosszúságú füzért fogad el, ha közte van az argumentumfüzére elemeinek. Persze azon a helyen, ahova a „láthatatlan kurzor" éppen mutat!

2. A NOTANY(str) függvény

Ezek után könnyű elképzelni, hogy a NOTANY függvény mit csinál. Eltalálta a kedves olvasó: bármilyen karaktert elfogad következő karakternek, hacsak nincs közötte az argumentumfüzérében felsorolt tiltott karaktereknek. „Következő karakteren" pedig azt értjük, hogy abban a pozícióban kell vizsgálatot folytatni, ahová a „láthatatlan kurzor" mutat.

3. Amikor nem kell az egyenlőségjel

Hogy használjuk akkor ezeket a függvényeket a betűk és a nem betűk szétválogatására? Például így:

Az „egykar" utasítás eltette a következő karaktert a „kar" változóba, ezt kell megvizsgálnunk. Ha előzőleg már az egész sorban kisbetűkkel cseréltük le a nagybetűket egy REPLACE utasítással, akkor már nagybetű nem lehet „kar"-ban. Válasszuk szét három részre a karaktereket: kisbetűk; szóközök vagy tabulátorjelek; egyéb jelek.

Ha a NOTANY függvényt alkalmazzuk először, argumentumaként a kisbetűkkel, akkor ez kisbetűk hatására ugrik a hibaágra (f), egyébként pedig a sikerágra (s). A következő vizsgálatnál ez utóbbi ágat választhatjuk szét például az ANY függvénnyel, ha argumentumaként a szóközből (" „) és a tabulátorjelből (CHAR(9)) képezünk jelsorozatot. Így elkerülhetjük, hogy minden létező jelet fel kelljen sorolnunk, azokat is, amelyek esetleg csak valami véletlen folytán kerültek bele a szövegfájlba.

A három utasítássor a következő lehet, ha — mondjuk — számolni akarjuk a nem üres jeleket:

```
      kar notany(hlabc)          :f(betu)
+             -                  s(ures)
ures kar any(" „ char(9))        :f(jel)
+                                s(egykar)
jel jel = jel + 1                :s(egykar)
```

A nagy dobás

A Snobol nagy dobása azonban még csak ezután következik. Van ugyanis egy olyan adattípusa, amely nagyon hasonlít a tömbhöz, bizonyos értelemben azonban jóval többet tud nála. Nevezetesen: füzérrel lehet indexelni! Most a kiszemezgetett betűinket nem kell tehát valami furfangos módon számindexekké alakítanunk (bár a Snobolban ez sem nehéz), hanem közvetlenül felhasználhatjuk a készítendő csodatömb elemeinek indexelésére.

Figyeljük meg a következő utasítássorokat:

```
b0 = table(35)
....
b0<kar> = b0<kar> + 1
```

Az első utasítás (valahol a program elején) nevet adott a csodatömbnek, és meghatározta a típusát. Ezenkívül azt is megmondta, hogy kezdetben 35 sora legyen, meggyorsítva ezzel az adminisztrációt. (Ha ezt nem írjuk oda, először 10 sort vett volna fel a program, majd 10-esével bővítette volna.)

A másik utasítás eggyel növeli minden végrehajtáskor a csodatömb „kar"-adik elemének tartalmát, ahol kar egy változó neve. De direktben is beleírhatnánk a csodatömbbe, akár ékezetes betűkkel is, például így:

```
b0<"érthetetlen"> = "hecsupászimuli"
```

Problematikus csak a csodatömb elemeinek visszaíratása lehet, különösen, ha nem tudjuk, hogy milyen indexekkel tettük el az elemeit. Hiszen nem kérdezhetjük meg, hogy például mi a harmadik eleme, mert ezt a nyelvet nem érti. Mindez azonban nem okoz gondot, ha közönséges tömbbé konvertáljuk:

```
b1 = convert(b0,"array")
```

Ettől az utasítástól számítva a b1 olyan kétdimenziós tömb, amelyből a szokásos módon elővehetjük csodatömbünknek a konvertálás pillanatában érvényes értékeit. Az első oszlopban jelennek meg az indexek, a másodikban az indexhez tartozó értékek.

Trükkös előrendezés

A csodatömb elemeinek rejtett rendezettségét kihasználhatjuk arra, hogy mi határozzuk meg ezt a sorrendet. Például előre biztosíthatjuk a magyar ábécének megfelelő sorrendet, hiszen a gyakoriság számlálása közben a Snobolnak közömbös az indexek sorrendje. Hogyan oldható ez meg?

Például úgy, hogy a magyar ábécé sorrendjének megfelelően 0-kat teszünk be minden számlálórekeszbe. A Snobol a beérkezés sorrendjében veszi föl az elemeket, ez pedig a későbbiekben nem változik meg.

Figyeljük meg a következő utasításokat:

```
     ahabc = hlabc
     b0 = table(35)
cikl habc len(1) . kar =         :f(olvas)
     b0<kar> = 0                 :(cikl)
```

Az első utasítás arra szolgál, hogy ne faljuk fel közben a hlabc füzért, hiszen arra később szükségünk lehet.

Ami a shareware változatból hiányzik...

Időnként bosszúságot jelent, hogy a shareware változat csak egész számokat ismer, és azoknál is elég alacsonyan van a korlát (32 K-nál). Némi ügyeskedéssel ki lehet játszani ezeket a korlátokat, de a program méretét kétségkívül megnövelik a betoldások. (Akinek gyakran van szüksége rájuk, függvényeket készíthet a korlátok kiküszöbölésére.)

További magyarázat helyett ajánljuk olvasóinknak, hogy használják egészséggel a mágneslemez mellékletünkön található gyakoriságszámláló programokat, és próbálják megérteni a logikáját. Az egyik program (betugyak.sno) a magyar ábécé sorrendjének megfelelően írja ki az eredményt, a másik egy kvázigyakorisági sorrendnek megfelelően, külön kezelve a magánhangzókat, külön a mássalhangzókat. A programok hívásakor kell megadnunk az inputfájl és az outputfájl nevét a parancssorban.

Például:

```
C:\SNO\>sno betugyak /I=arany.001
/O=arany1
```

vagy

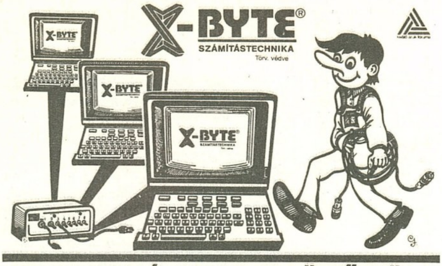
```
C:\SNO\>sno beturang /I=babits.003
/O=babits3
```

Itt nem engedi meg a program a szóközöket az egyenlőségjel két oldalán, kellemetlen, hogy az NC (Ctrl-Enter) pedig csak úgy másol fájlnevet, ha egy pozícióval tovább léptetjük a kurzort. Ilyenkor utólag töröljük ki a fölösleges szóközöket. A másik kényelmetlenség: az /I-t és /O-t nagybetűkkel kell írni.

Olvasóink nyelvi vizsgálódásainak megkönnyítése céljából (és a későbbi feladatokra is gondolva) mágneslemez mellékletünkön elhelyeztünk egy csokorra való verset/versrészletet ismert és kevésbé ismert költőink ismert és kevésbé ismert műveiből, műfordításaiból. Külön listákban adjuk meg a szerzőket, és külön a művek címét. Kérdés: hogyan tartoznak egymáshoz a szerzők, a versek és a címek? Melyek eredeti művek, és melyek műfordítások? A legjobb megfejtőknek könyvjutalommal kedveskedünk.

Várjuk olvasóink ötleteit, javaslatait (sőt lassacskán már elkészülő programjaikat is), hogy feltérképezzük, milyen feladatok megoldására lehetne ügyesen felhasználni a Snobolt. Reméljük, sokat tanulhatunk még egymástól.

Vargha Dénes



KAPCSOLÓDJON A JÖVŐHÖZ!

SZÁMÍTÓGÉPHÁLÓZATOK

MILYEN TÍPUSÚ HÁLÓZAT SZÜKSÉGES ÖNNEK?

ARCNET, ETHERNET, RS 232,
IBM CABLING SYSTEM AT & T,
SYSTIMAX, ÜVEGSZÁL,
NOVELL?

JÖJJÖN EL HOZZÁNK!

1138 Budapest, Népfürdő u. 17/e. Telefon: 173-1329 Fax: 173-1530

Egy kávé és üdítő mellett
segítünk a választásban.

CSÖKKENTETT ÁRAK, VÁLTOZATLAN MINŐSÉG!

INFORMÁCIÓKÉRÉS: 57 ▼

ÚJDONSÁG !

- Sekonic CTS-8 plotter 69.000,-Ft+ÁFA
  - A3 méret
  - hat szín
  - HP kompatibilis
- S3 XGA videokártya 21.680,-Ft+ÁFA
  - háromszoros sebesség

Lokál bus-os alkatrészek :

- S3 XGA kártya 26.880,-Ft+ÁFA
- SCSI kontroller 14.980,-Ft+ÁFA
- 486/33 alaplap 68.000,-Ft+ÁFA
  - 256 KByte cache

WD 280 winchester (80 MB) 23.980,-Ft+ÁFA

WD 2120 winchester (120 MB) 28.000,-Ft+ÁFA

Hálózatépítés, karbantartás .

DATA DOCTOR
SZÁMÍTÁSTECHNIKAI KFT

BUDAPEST 1073. Erzsébet krt. 25-27. I.em.9.
Telefon / Fax : 121 - 03 - 21

INFORMÁCIÓKÉRÉS: 14 ▼

HOKTRADE Co. Ltd.

INDUSTRIAL AND COMMERCIAL CO. LTD.

NOTEBOOK SHOP

Notebookok, 286, 386SX, 386DX
Kodak, CITIZEN Notebook-printerek
Pocket-modemek, pocketfax-modemek
Akkumulátorok, autóadapterek
Kiegészítők nagy választékban

PC CENTER SHOP

Komplett konfigurációk
Alkatrészek töredékáron
Tartozékok nagy választékban
Használt gépek árengedménnyel

Minden, amire Önnek szüksége lehet!

HOKTRADE Kft.

1012 Budapest, Attila út 93.
Tel: 202-4166 Fax: 175-0446

INFORMÁCIÓKÉRÉS: 27 ▼

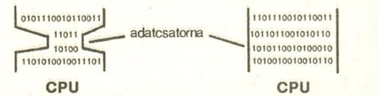
SPECTRAL Kft.

1145 Budapest, Amerikai út 39.
Tel./Fax: (1)-183-7015

Újdonság a GIGABYTE-tól

Egy új fogalom a számítástechnikában: ***LOCAL-BUS***
Csak egy pillantást kell vetnie az ábránkra

Hagyományos ISA-BUS	LOCAL-BUS
8/16 bit 8 MHz	32 bit, proc. sebesség
DISPLAY, HDD	DISPLAY, HDD



Így már Ön is tudja, miért nyújtanak többszörös teljesítményt *(az ISA alaplapúakkal azonos árszintenl)* azok a számítógépek, amelyekben ***LOCAL-BUS*** is található. Ha Ön a pénzéért a legnagyobb teljesítményű gépet akarja kapni, hívjon fel bennünket és mi bemutatjuk, milyen is egy gyors rendszer a valóságban!

A NOTEBOOK-PIACON EGYEDÜLÁLLÓ ÚJDONSÁG:

CHAPLET

386SL, 25 MHz, beépített TRACK-BALL-lal, SR FAX-szal, 10" MVGA, 60–120 MB HDD + DOS 5.0 + WINDOWS 3.1, kell még egy hordozható lapadagolós 24 tűs nyomtató és megvan a mozgó irodája egy diplomatatáskába építve, a ***MIKROPAKK***.
Nem fér el az asztalán? BOOK SIZE számítógépet ajánlunk, a

DYNA STATION-t.

Munkaállomásnak is ideális – 386SX, 25 MHz, 1,44 FDD, LAN csatoló, 8/16 bit.

INFORMÁCIÓKÉRÉS: 48 ▼

Campbell, Joe:
C tippek és trükkök.

Budapest, 1992.
Novotrade Kiadó Kft. —
Prentice Hall, 457 oldal.
Ára: 860,– Ft.

Mivel a C nyelvet legtöbbször rendszerközeli, a hardver képességeit messzemenően kihasználó programok írásánál használják, a programozók előbb-utóbb szembekerülnek azzal a problémával, hogy a program egyes függvényeit assembly nyelven célszerű megírni.

Az ilyen függvények írása az operációs rendszeren és az assembly nyelven kívül a C nyelvű programok működésének alapos ismeretét igényli. Ezen ismeretek elsajátításában nyújthat nagy segítséget ez a könyv. Kezdőknek és más gépi környezetből jött profiknak egyaránt értékes a könyvben található anyag, mely felöleli szinte az összes idevágó ismeretet, példaprogramokon keresztül bemutatva az ilyen programok fejlesztésének menetét a rendszerprogramozás különböző területein. Igen kellemes, hogy a magyarázaton és a példaprogramokon kívül az adott programban bevezetett új ismereteket külön kiemeli a szerző, valamint hogy az egyes függvények előtti rövid leírás mindig tartalmazza a használatukhoz szükséges információkat. Szintén kellemes meglepetés az irodalomjegyzék, ez ugyanis a felhasznált művek címein kívül tartalmazza rövid leírásukat is, ezáltal segítve az elmélyülést az egyes területeken.

A könyv hiányosságai közé kell sorolni, hogy a tartalomjegyzék valószínűleg igen „átszellemült" alkotás (mivelhogy az utolsó négy oldal üres — mutatva: íme, itt kellene lennie!), valamint hogy a DOS függvényei szám szerint fel vannak ugyan sorolva, de a paraméterezésük nélkül.

Goda Attila

Bibliográfia

Összeállításunkban ez alkalommal is olyan könyvek között válogattunk, melyek rovatunkban még nem szerepeltek. Az évszám megjelölése nélkül kiadott LSI-könyvek 1992-ben kerültek forgalomba.

Abaffy József: Bevezetés a Turbo Pascal 5.0-ba (kézirat). Budapest, 1990. Aula Kiadó, 132 oldal. Ára: 180,– Ft.

Abaffy József — Tarlós Béla: A programozás alapjai. Budapest, 1991. Aula Kiadó, 155 oldal. Ára: 271,– Ft.

Abonyi Zsolt: PC hardver kézikönyv. Budapest, 1992. ComputerBooks, 276 oldal. Ára: 549,– Ft.

Balogh János—Dr. Dedinszky Ferenc: FoxPro 2.0. Budapest, 1992. ComputerBooks, 466 oldal. Ára: 695,– Ft.

Bárdos Attila—Lőcs Gyula—Rácz Lajos—Sályi János: A programozás alapjai (kézirat). Budapest, 1991. Számalk, 206 oldal. Ára: 390,– Ft.

Csépai János: Számítástechnikai alapismeretek (kézirat). Budapest, 1991. Aula Kiadó, 109 oldal. Ára: 224,– Ft.

Csetényi Artúr — Mócsai Andrea: Szövegszerkesztés ChiWriterrel.
Ormós Zsoltné: DBase III Plus.
Gáspár Bencéné: Bevezetés a Quattro táblázatkezelő használatába.

Csépai János: Az MS DOS operációs rendszer áttekintése. Budapest, é. n. Aula Kiadó, 36 + 33 + 40 + 7 oldal. Ára: 221,– Ft.

Cserhalmi Zsolt: WINDOWS 3.1. Alapok, újdonságok, programismertetés, tippek és trükkök. Budapest, 1992. Computer Panoráma Kft., 126 oldal. Ára: 550,– Ft.

Cyffka, Günther: DR DOS 6.0 Quick & Easy. Budapest, 1992. Mikroszerviz Rt. — Berger Kft. 137 oldal. Ára: 690,- Ft.

Darab Tamás: A gépesített értelem. Vázlatok a mesterséges intelligencia filozófiájáról. Budapest, 1991. Áron László, 110 oldal. Ára: 117,– Ft.

Dr. Dedinszky Ferenc: Clipper 5 (5.0, 5.01 és segédprogramjai). Budapest, 1992. ComputerBooks, 481 oldal. Ára: 699,– Ft.

Detrik Péter: Az SQL nyelvről. (Alaplap Füzetek). Budapest, 1992. Cédrus Kiadó, 94 oldal. Ára: 375,– Ft.

F. Ható Katalin: MS Works 2.0 felhasználóknak. Budapest, 1992. ComputerBooks, 121 oldal. Ára: 398,– Ft.

Gábor András: Számítógépes információrendszerek (kézirat). Budapest, 1990. Aula Kiadó, 143 oldal. Ára: 202,– Ft.

Hargittai Péter: Harward Graphics. Budapest, é. n. LSI Oktatóközpont Alapítvány, 417 oldal. Ára: 590,– Ft.

Honti József — Honti Mónika: Közérthetően a CorelDRAW-ról — az alapoktól a 2.0-ig. Budapest, é. n. LSI Oktatóközpont, 186 oldal. Ára: 399,– Ft.

Jamsa, Kris: Egyszerűen DOS. Budapest, 1992. Panem — McGraw-Hill, 181 oldal. Ára: 179,– Ft.

Kaszanyiczki László: DR DOS 6.0 kapcsolat a Windows 3.0-val. Budapest, é. n. LSI Oktatóközpont Alapítvány, 304 oldal. Ára: 490,– Ft.

Kelemenné Ternai Katalin: A C programnyelv. Budapest, 1991. Aula Kiadó, 104 oldal. Ára: 300,– Ft.

Kenczler Mihály: CorelDRAW! 2.0. Budapest, 1992. ComputerBooks, 100 oldal. Ára: 330,– Ft.

Kernighan, B. W.—Ike, R.: A Unix operációs rendszer (2. kiadás). Budapest, 1992. Műszaki Könyvkiadó, 362 oldal. Ára: 687,– Ft.

Kis János — Szegedi Imre: Vírushatározó. (Alaplap Könyvek 4.). Budapest, 1992. Cédrus Kiadó, 283 oldal. Ára: 256,– Ft.

Klucs László—Koleszár Gyula: Excel for Windows 3.0. Táblázatkezelő Windows alatt. Budapest, 1992. LSI Oktatóközpont, 226 oldal. Ára: 443,– Ft.

Martin, James—Chapman, Kathleen: Lokális hálózatok. Budapest, 1992. Novotrade—Prentice Hall, 367 oldal. Ára: 890,– Ft.

Molnár Mátyás: WORD 5.5. Budapest, 1992. ComputerBooks, 247 oldal. Ára: 388,– Ft.

Mörk Péter: Word for Windows 2.0. Budapest, é. n. LSI Oktatóközpont Alapítvány, 143 oldal. Ára: 290,– Ft.

Norton, Peter: Az IBM PC programozása (2. kiadás). Budapest, 1992. Műszaki Könyvkiadó, 371 oldal. Ára: 495,– Ft.

Nyéki Lajos — Nagy Tamás: Turbo Basic. Budapest, 1991. LSI Oktatóközpont, 385 oldal. Ára: 453,– Ft.

Pirkó József: Turbo Pascal 6.0 & for Windows programozási kézikönyv. Budapest, é. n. LSI Oktatóközpont, 682 oldal. Ára: 799,– Ft.

Szabadhegyi Csaba: A Word 5.5 használata. Budapest, 1992. Reál (Múzsák Közművelődési Kiadó), 130 oldal. Ára: 240,– Ft.

Személyi számítógépek és tartozékainak felépítése és működése. Budapest, 1991. Trafcomp Kft., 107 oldal. Ára: 298,– Ft.

A Mikrobazár rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk.
A kereskedelmi tevékenységet szolgáló apróhirdetések tarifája gépelt soronként (60 karakterenként) 300 Ft. Kérjük, hogy a hirdetés díját a Cédrus Kiadó Kft-nek az Általános Értékforgalmi Banknál vezetett 204-19417 számú számlájára utalják át, vagy postautalványon fizessék be a Cédrus Kiadó Kft címére (1441 Budapest VIII., Reguly Antal u. 8.), a hátoldalon feltüntetve, hogy apróhirdetés. A befizetést igazoló szelvényt a közlendő hirdetési szöveggel együtt az Alaplap szerkesztőségéhez (a kiadóéval azonos címre) küldjék el.
A nem kereskedelmi célú egyéni hirdetések közlése INGYENES!

ELAD

Enterprise programok eladók. Válaszboríték ellenében listát küldök. 2000 program, sok kedvezmény, ajándék. Cím: Zemen László, 1164 Budapest XVI., Olló u. 16.

Eladó egyben vagy részenként **Commodore 128** 1571-es floppymeghajtó, magnó, 12"-os zöld monitor, 1531-es egér, 2 db joystick, programok. Cím: Gonda László, 5130 Jászapáti, Attila u. 3.

PC videovezérlőkártyák (Hercules, CGA, EGA, VGA) hardveres ékezetesítése CWI vagy egyéb tetszőleges kódkészlet szerint. PC Turbo klubtagoknak kedvezmény! Cím: Lóth Tamás, 1558 Budapest, Pf. 76.

Számítástechnikai oktatás IBM PC gépen bármilyen témában a legolcsóbban! Beszerzési tanácsadást és programkészítést is vállalok! PC Turbo klubtagoknak kedvezmény! Cím: Fridl György, 1116 Budapest XI., Szalóki u. 6.

Programokra, segítségre, cseretársakra van szükséged? Az ASIS megoldja problémáidat! Bárhol laksz, bármilyen géped van, írj! Kérésre ingyenes tájékoztatót küldünk. Cím: ASIS, 1425 Budapest, Pf. 729. Tel.: 142-8075.

Kedves Szülők! Miért csak a gyermekük ért a számítógéphez? Tanulja meg Ön is a Basic vagy más ismert programnyelvet a legelterjedtebb hazai gépeken! Cím: Oszlányi Norbert, 8300 Tapolca, Juhász Gyula u. 47. B/15.

VESZ

Egyetemi hallgató kér segítséget ahhoz, hogy juttassák el címére **fölösleges vagy üzemképtelen hardvereszközeiket** (részegységeiket). Cím: Dömsödi Mihály, 2038 Sóskút, Orgona u. 11.

Keresek IPX, SPX, NET5 és NETBIOS programozói leírásokat. Cím: Somlai Gábor, 6000 Kecskemét, Széchenyi krt. 46. Tel.: (76) 23-630.

Keresek Headstart Explorer XT-hez **256 kB RAM PAC memóriabővítőt.** Cím: Nagy Sándor, 8200 Veszprém, Lóczy u. 20/a.

CSERÉL

Keresem a Microsoft **Macro Assembler 6.0**-ás változatát, de az 5.1 vagy e fölötti verzió is érdekel. Keresem továbbá a Borland C++, a Microsoft Quick C és a Turbo Pascal 6.0 programokat. Csereajánlataim az alábbiak: MS Windows 3.1, Norton Desktop for Windows, MS Word for Windows 2.0, MS Excel 3.0 for Windows, MS Works 2.0 for Windows, Superbase IV. for Windows, HDC First Apds for Windows, Chessmaster for Windows. Cím: Lukácsy Gergely, 1118 Budapest XI., Törökugrató u. 4. I./1. Tel.: 173-5044.

IBM PC XT/AT játék- és felhasználói programokat cserélnék. Felbélyegzett válaszboríték ellenében listát küldök. Cím: Révai Csaba, 2316 Tököl, Huba u. 36.

Cserélnék **SoundBlaster FM zenéket** (.ROL, .CMF, .MID). Cím: Szekrényesi Zsolt, 3535 Miskolc, Endrődi út 24.

A DOS és az alatta futó rendszerprogramok...

SzabaDOS, LINK magatartás

Alábbi beszámolónk lényegében kész volt, amikor megjelent az Alaplap augusztusi száma. A 11. oldali kérdést tehát mellesleg és tulajdonképpen már megválaszoltuk, de célszerűnek látjuk meg is magyarázni. Nos, a választott „.EXP" formát azért használjuk, mert így egyértelmű. Mit sugall az „exp program" kifejezés? Nyilvánvaló — gondolhatnánk —, az exponenciális függvénnyel van kapcsolatban. Ezzel szemben a Lahey Fortran által létrehozott, a 640 K korlátot nem ismerő, futtatható, „.exp" kiterjesztésű programról van szó, amit azért írunk „.EXP" formában, mert a DOS is csak nagybetűt használ.

Idézetek az ALAPLAP 1992. júliusi számából:

„Mi is az, hogy operációs rendszer? Egy szükséges feltétele a hardver munkára foghatóságának."

„A DOS nélkülözhetetlen ahhoz, hogy a számítógépet egyáltalán használni tudjuk."

„Enélkül nem megy."

„...az ügyes felhasználói programok elrejtik előlünk..."

„MS-DOS 5.0 — Végre egy kiforrott, nem elkapkodott, átgondolt DOS-változat."

Egy olyan, PC-vezérelt környezetellenőrző mérőrendszer létrehozása volt a feladatunk, amelynek a „megengedett" kiesési ideje évi egy óra (a 8760-ból). Ez természetesen azzal járt, hogy a programot „bolondbiztos" kivitelben kellett elkészítenünk.

Anélkül, hogy a részletekbe belemennénk, két szokatlan követelményt említünk csak meg. Egyrészt hatástalanítani kellett a Pause billentyűt és a Ctrl-Alt-Del kombinációt, hiszen ha ez(eke)t véletlenül leütik, a mérés meghal, másrészt pedig csak kétkezes adatbeadás fogadható el (ha a billentyűzetre rátesznek egy könyvet, az „érintett" karakter folyamatosan bemegy a pufferba), például A helyett Ctrl-A, ami végül egy új billentyűzetkezelő (keyboard driver) program írását tette szükségessé.

Utóbb kiderült, hogy ez volt a feladat legkönnyebb része, mivel ez teljes egészében saját fejlesztés volt. Nehézségeink a látszólag készen rendelkezésre álló rendszerrutinok meghívása esetén támadtak.

Kellemetlen tapasztalatainkat mások okulására — elrettentésére? — ezúton tesszük közzé.

Csalódni nem muszáj!

Persze a „legkönnyebb", bár felsőfok, kevesebbet jelent, mint az alapfokú „könnyű". A billentyűzetkezelő rutin szépen átveszi a BIOS 09H megszakításkezelésének feladatait: a Shift, Ctrl, Alt és a CapsLock, ScrollLock, NumLock billentyűk kezelésének eseményeit nyilvántartja, majd a nyilvántartás alapján a többi billentyű helyére jellemző scan code értékéből (ez a szám az, amit a CPU megkap) kikeveri a billentyű ASCII-kódját. Nos, a QWERTY-effektus miatt ehhez szükség van egy kis táblázatkezelésre is: a billentyűzet Q-sorának, A-sorának, Z-sorának azonosítása a sor bal szélső billentyűjének scan code-jával történhet meg.

(Ha valaki nem ismerné: az angol írógép-billentyűzet elrendezése ergonómiailag igen rossz — hát még a magyaré! Mégis, ezt tanulták meg a vakon író felhasználók. Miattuk viszont nem lehet eltérni a QWERTY-től...)

Az assembly nyelven írt rutint a Microsoft MASM szépen, szabályosan le is fordította. Ez a rész valami ilyesmi (AX-ben a beolvasott érték):

```
CMP AX,SCANQ
JL SZAMJEGYSOR
CMP AX,SCANA
JL QBETUSOR
CMP AX,SCANZ
JL ABETUSOR
ZBETUSOR: ; a legalsó
billentyűsor feldolgozása
JMP TOVABB
SZAMJEGYSOR: ; hasonlóan a
legfelső sorra stb.
```

Mivel a ZBETUSOR, SZAMJEGYSOR stb. címkék értéke csak a program betöltésekor lesz ismert, a LINK szerkesztőprogram feladata az .EXE fájl megfelelő kialakítása. Megdöbbentő volt az eredmény: a DEBUG listájából kiderült, hogy a JL SZAMJEGYSOR, a JL QBETUSOR és a JL ABETUSOR utasításoknak a LINK által adott címei közül egy mindig, néha pedig kettő is hibás volt, azaz a futó program végrehajtása során a vezérlés például egy 3 bájtos utasítás középső bájtjára adódott át, az ott talált memóriaszemetet utasításként végrehajtva. (És mivel éppen a billentyűzetkezelés volt a feladat, a DEBUG a futtatáshoz már használhatatlan volt...) Az, hogy hány ugróutasítás volt hibás, attól függött, hogy a fenti programrészlet előtt milyen hosszú szöveg állt. (A hibakereséshez beszúrva egy üzenetet kiíró utasítást — hosszabb előzmény — az alcímzés máris máshová mutatott.)

LINKségek

A LINK okozta gondokat illusztrálja az az (átrendezett) memóriatérkép, amelyet a mágneslemezen lehet tanulmányozni. Az assembly nyelvű programok megírása túl munkaigényes, ezért a programban az egyszerűség kedvéért vannak C és Fortran rutinok is: a mérés kiértékeléséhez szükséges integrálást botorság lenne nem Fortranban programozni. Az alkalmazott három Microsoft fordítóprogram által létrehozott .OBJ modulokat össze lehet szerkeszteni. De...

Attól függően, hogy a LINK-nek milyen sorrendben adtuk meg a két runtime könyvtár nevét (LLIBFOR7, LLIBC7), amelyeket a fordítóprogramokkal kapott SETUP programok kompatibilisként hoztak létre, a két .EXE fájl mérete, szerkezete eltérő lett. A könyvtári rutinok „SEGMENT BYTE PUBLIC 'CODE'" deklarációja indokolt, hiszen így lesz a program a legkisebb. Sajnálatos, hogy ennek következtében a paragrafushatárra igazítás teljesen elmarad. Az Intel 80x86 CPU-k címzési módja közismert: a paragrafushatár címét 10H-val osztva kapjuk meg a szegmensregiszter tartalmát, a címregiszterben pedig a cím és a paragrafushatár címe közötti különbség (offset) van. Esetünkben a 8-asra végződő kezdőcím 8-asa elveszett, de mivel az osztás utáni csonkítás véletlenül jó szegmensregiszter-tartalmat adott ki, a program elindult. Megállni már nem tudott: vagy azonnal a futtatás után (a DOS halála miatt), vagy később, de mindenképpen rendszerösszeomláshoz vezetett. (Rendszeresen kaptunk ilyen üzeneteket: „Drive C: is not ready", avagy a C:-re vonatkoztatva: „not bootable". Ezek után a FAT manuális rendbetétele következett.)

(A Fortran programok R6001 hibaüzenetének — „null pointer assignment" — magyarázata a Microsoft-dokumentációban 15 sor. Ennek lényege, hogy a program akár jó eredményeket produkálva futhat is, de a benne lévő potenciálisan súlyos hiba — nem a program saját adatterületére írás — miatt a jövőben és/vagy más környezetben, például a rutint másik főprogramból híva bármikor meghalhat.)

Valószínűsítjük (bár bizonyítani nem sikerült), hogy az előzőleg említett QWERTY-címzési gondok is ilyen eredetűek voltak. Ez lehet az oka például annak is, hogy egy méréskiértékelő programunk, amelyet forrásnyelven adtunk át felhasználásra, a DOS Hollandiába szánt változata alatt működésképtelen. (Mellesleg, szakterületünkön az USA-ból beszerezhető programok vagy a Ryan-McFarland, vagy a Lahey fordítóprogrammal, azok saját szerkesztőjével készülnek, a Microsoft a kínálatban alig fordul elő.)

Ezek után felmerül a kérdés: miért használjuk mégis ezt az operációs rendszert?

Miért, nos miért?

Mérőrendszerünkre az jellemző, hogy a mérési eredményeket a teljes környezetvédelmi mérőrendszer központi gépének adja át; a felhasználók azzal a géppel vannak kapcsolatban. Ez a gép egy DEC gyártmányú MicroVAX. (PC-nk DECstation200 típusú.) A hálózati kapcsolat a DECneten (Ethernet) valósul meg, amit a PC-n installált DECnet DOS, a Microsoft DOS 3.30 megfejelése tesz lehetővé. A hálózati rutinok Microsoft C forrásnyelven állnak rendelkezésre. Logikusnak látszott ezt a szoftverkörnyezetet használni.

Természetesen a teljes rendszer jogtiszta, „hófehér", azaz nincs rajta semmiféle szeplő, amit a „kalózok" ejtettek volna rá.

A MicroVAX programját mások írták. Koncepciójuk szerint az a gép megszólíthatja a miénket, de fordítva tilos. Ha tehát valami rendelleneset találunk, csak helyi riasztást adhatunk ki. Azért, hogy legalább ezt üzembiztosan kiadhassuk, korlátoztuk az operátorok géphez fordulási idejét: a parancsok beadásának kezdetétől végéig (a parancstól függően) csak 3—9 másodpercük van. Ezt az időt a PC saját órájával, az 1CH megszakítást kezelve akartuk mérni. (Ez a megszakítás a PC órájának — timerének — a felhasználók részére fenntartott megszakításmásolata, 55 ms-onként adódik ki. Ha tehát egy 3 másodperces parancsot várunk, 3/0,055 = 54 ilyen megszakítást kell leszámlálnunk, miközben a főprogram bármi mást csinálhat. Az 54 elérésekor kell törölni az operátori parancs fogadását.) Mindaddig, amíg PC-nk offline, majd a teszteléshez létrehozott kis Ethernet hálózaton működött, a program is működött. Bezzeg a környezetellenőrző mérőrendszer DECnetjére kapcsoláskor...

Sikerült pillanatokon belül szétverni a MicroVAX-ból, annak nem PC termináljaiból, a kapcsolt PC-kből stb. álló rendszert. A diagnózis ez volt: ha a mi programunk kezeli az 1CH-t, a DECnet és a MicroVAX is meghal, még csak a lokális parancsok sem működnek. Tüneti kezelésként kivettük a programból ezt a „user's timer driver"-t, mire minden rendbejött. Mivel a DECnet DOS dokumentációjából (jogtiszta!) nem derül ki, hogyan is működik a hálózati kártyával kapott DECNET.SYS, megint csak valószínű magyarázatot találtunk: a DEC hálózati kártyáján lévő BIOS-kiegészítés saját céljaira használhatja a felhasználónak szabad garázdálkodásra felkínált 1CH megszakítást. Azzal, hogy „elvettük" ezt tőle, PC-nk a bejelentkezett felhasználók közül az 1CH-t kezelő rutin futásának idejére mindig kitörlődött (másodpercenként 18-szor), így aztán semmiféle adatátvitelt nem lehetett a rendszer nem PC termináljairól végrehajtani (a PC-kről viszont igen).

Mi mást tehettünk, a PC-be betett külön mérésvezérlő kártyán amúgy is meglévő órát kezdtük használni. Ez nem ad be megszakításkérelmet, tehát folyamatosan figyelnünk kell rá. Programunk így sokat veszített eleganciájából, s ezzel megbízhatóságából is. (Az újonnan használatba vett hardvereszköz új hibaforrás.)

A hálózati kártyán van az egér konnektora is. Az egérrel sok bajunk volt. Időnként behavazódott az egész, a mért adatokat interpretáló rajz, ha az egeret balszerencsés pillanatban mozdítottuk meg. Ennek okát sikerült meglelni: ha az egér hardvere nem figyeli a monitor képváltási idejét, ez történhet. De miért? A leírás szerint az egér úgy viselkedik, mint a Microsoft egere. Ezzel szemben a DECMOUSE.SYS-ben ezt találtuk: „Ez egy LOGITECH egér. Az egérkezelő programok általában a 'Copyright Microsoft' szöveget keresik." Ezzel a korrekt kijelentéssel a DEC egere Microsoft-egérré változott, anélkül, hogy a hardver viselkedése igazodna a Microsoft-egértől elvárható viselkedéshez.

A DOS Technical Reference Manualje pontosan leírja a rendszer szolgáltatásait. Ilyen például az Equipment List ellenőrzése, azaz meg lehet tudni, milyen hardverelemeket tartalmaz a PC. Ezzel a kísérlettel is kudarcot vallottunk: a DECnet DOS-t megalapozó DECstation200-BIOS táblázatának ellenőrző összege nem képezhető a komponens adatokból. Még szerencse, hogy a BIOS megszakítási címeit megtartották...

A mért adatokat archiválni is kell. Ha a program felismeri, hogy nincs rajta a hálózaton, hajlékonylemezre menti ki az eredményeket. Az USA-ban az állami szabványban előírt tulajdonságú Fortran akkor, ha az OPEN utasításban szerepel az IOSTAT= paraméter, a fájlmegnyitás hibái esetén (például nincs lemez a meghajtóban) az ott megadott nevű változónak adná át a hibakód értékét, ha nem szólna közbe a Microsoft DOS-programozóinak szelleme. Ők ugyanis csak azt tudták elképzelni, hogy a felhasználó vigyázó szemeit a képernyőre függeszti (évi 8759 órán át), arra várva, hogy tudassák vele, mit tegyen a rendszer kiszolgálása érdekében. Így aztán az a program, amelyiknek felügyelet nélkül kell működnie, nem használhatja ki ezt a kényelmes megoldást. A Microsoft Fortran kézikönyv a szabványos működést írja le,

amit aztán a Microsoft DOS lehetetlenné tesz: a BIOS felismeri, hogy a meghajtó üres, a DOS rutin pedig ahelyett, hogy a Fortran programmal közölné az IOSTAT= (mondjuk) 13 értéket, a tényről üzenetet küld a standard error outputra, azaz a képernyőre. Ezt az üzenetet nem lehet letiltani, átirányítani. A DOS rutinja ezután addig vár, míg a felhasználó az eredeti BIOS billentyűzetmeghajtó útján értesítést nem ad a DOS-nak a hiba elhárításáról. Ezt követi az OPEN végrehajtása, az IOSTAT=0 értékadással (ha nincs más hiba).

A megoldás: el kell vetni a Fortran és C nyelvekben, valamint a DOS-tól (a 21H megszakítással) rendelkezésre álló összes ajánlatot, helyettük a BIOS hajlékonylemez-kezelő rutinjait kell meghívni. A DOS-hoz kizárólag akkor szabad fordulni, ha a BIOS válasza: a lemez O.K.

Ahogy ment, ahogy nem

Maga az adatmentés a DOS meghívásával történik, a C nyelv SYSTEM függvényének aktivizálásával. Ez megnyitja a másodlagos DOS-környezetet, annak COMMAND.COM-ja értelmezi az argumentumként beírt parancsot stb. A függvény értéke 0, ha a feladat sikerült, ha nem, akkor –1. Ez tesztelhető, például így:

```
I=SYSTEM('MOVE *.*
A: >NUL'//CHAR(0))
IF (I.NE.0) THEN
...
ELSE
...
END IF
```

Az argumentumban a „>NUL" jelsorozat célja, hogy a DOS válaszát ne engedjük be a rajz közepébe.

Az ebben a példában szereplő MOVE parancs valóban ismeretlen a DOS 3.30-ban és 5.00-ban, de egy már selejtezett XT-nk Microsoft DOS 3.10-ében megtaláltuk. (A „single user, single machine" elvet nem sérti, hogy az újabb DOS-t kiegészítettük ezzel az elemmel.)

A parancs gyors és megbízható, hiszen akkor, ha a forrás és a cél ugyanaz a meghajtó, nem másol mást, csak a directorybejegyzést.

Mivel a mérésvezérlés, archiválás, operátori parancsok fogadása a konzolról és az egértől stb. sokrétű feladat, programunk is nagyra nőtt. Előfordult, hogy valamelyik I=SYSTEM utasítás végrehajtása során a szabad memória túl nagy volt ahhoz, hogy a rendszer ne kezdje el a DOS parancsot, de túl kicsi volt a befejezéshez. Így aztán exlex állapot alakult ki, amin csak a gép kikapcsolása segített (nincs rajta Reset gomb).

A hologram nem talizmán...

A DOS 5.00 kézikönyvén ékesség az a hologram, amelyen kb. 25-ször olvasható a „Microsoft (R)" felirat. A fülszöveg szerint csak az a valódi DOS 5.00, amelyiken ez a hologram megvan. (Rendkívül szerényen nem az original, hanem a genuine szót használják, amely persze összetéveszthetetlen a geniusszal.) Nos, ennek hasznos tanácsa (a 280. oldal utolsó két sora), hogy használjuk a HIMEM memóriakezelő programot, így nagyobb területünk lesz a saját program számára. Kipróbáltuk. Áttértünk a Microsoft DOS 5.00-ra, vállalva a DECnet-féle DOS-szal kapcsolatos kockázatokat is. Az I=SYSTEM használata ugyanúgy végzetessé vált, mint a 3.30 alatt. Utóbb kiderült, miért. A CONFIG.SYS-ben akár benne volt a következő két sor:

```
DEVICE=HYMEM.SYS
DOS=HIGH
```

akár nem, a MEM parancs ugyanazt a memóriatérképet írta ki. Az ok ismeretlen maradt. Vissza is tértünk a 3.30-hoz.

Tanulságként azt szűrtük le, hogy jóllehet a DEC elismert számítógépgyártó cég, a Microsoft elismert szoftverfejlesztő, a kettő együttműködése túl sok megoldatlan problémát okozott ahhoz, hogy a Microsoft-termékek belső ellentmondásait egy vállrándítással elintézhettük volna. Ezért kényes feladatokhoz a jövőben csak egy kézből származó hardver-szoftver együttest fogunk alkalmazni, minden bizonnyal eredeti IBM-rendszert.

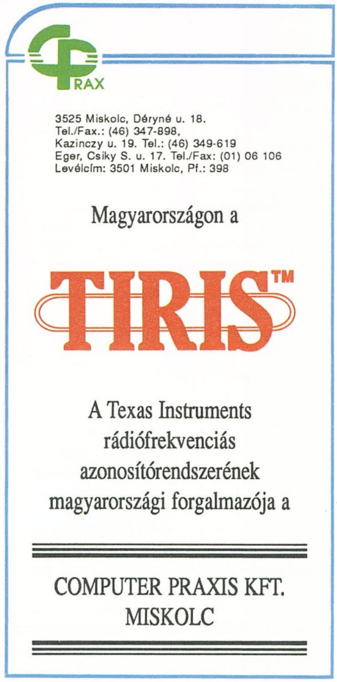
A „nagy kék óriás" más gépeivel szerzett két évtizedes tapasztalataink között voltak kellemetlenek is, de ilyen rosszak soha.

Egyébként a Microsoft LINK 5.03 azóta is produkált látványos eredményeket.

Szondi Egon János — Balázs László



INFORMÁCIÓKÉRÉS: 50 ▲



INFORMÁCIÓKÉRÉS: 10 ▲



INFORMÁCIÓKÉRÉS: 45 ▲

Takarékoskodj, okosan!

Összeállításunkban középponti helyen a takarékosság áll: hely-, idő- és költségtakarékos az új, 21 MB (!) kapacitású hajlékony mágneslemez (végre drive-ostul), beruházástakarékos a CAD programokat oktató kabinet, és sok-sok azonnali, igazi pénzt takaríthatunk meg a számítógépes csődfigyelés révén, midőn megtudjuk, ki nem volt elég takarékos — vagy elég óvatos.

Helytakarékos előnyök

A „kinőtt" winchester, a „disk full" üzenet egyidős a tárolóeszközök megjelenésével. Pedig szoftverre igenis szükség van. Csakhogy hová tegyük? A kérdés megválaszolásához kínál vadonatúj alternatívát a 3M. A gépkinövők zömének ma még nincs pénze állandóan cserélgetni nagyobbra vagy cserélhetőre a winchestert, hogy minden elférjen rajta, marad tehát a régi módszer: csak a legfontosabb adatainkat és szoftvereinket hagyjuk a winchesteren, a többit lementjük floppyk tömegére. Egy komolyabb szoftver így például 10-15 lemezen fér csak el. Ezen a kényelmetlen helyzeten segít az az új, nagy kapacitású hajlékony mágneslemez, amelyet — a világpremierrel közel azonos időben — a 3M Hungária Kft. forgalmazásában ismerhettünk meg.

Az újonnan fejlesztett 3,5"-os, 21 MB kapacitású Floptical diskette majdnem 14-szer több adatot tárol, mint a hagyományos 3,5"-os, 1,44 MB-os floppy. Használatával megoldódhat a nagy tömegű adatok tárolása olcsó, cserélhető mágneses adathordozón. Az új termék grafikus alkalmazásoknál, nagyméretű adatbázisok kezelésénél, kiadványok szerkesztésénél, szoftverek terjesztésénél, multimédia-előadásoknál, valamint adatok mentésénél jelent elsősorban ideális megoldást. Ez utóbbi esetben például egy teljes, 80 MB kapacitású merevlemez adatállományát 4 darab Floptical diszken tárolhatjuk, míg jelenleg ugyanerre a célra 50 darab 1,44 MB-os floppyra van szükségünk. De nemcsak a mentésre használt lemezek száma csökken, hanem a mentés ideje is jelentősen lerövidül, hiszen 21 MB adatállomány mentése a merevlemezről mindössze 3 és fél percig tart.

A Floptical diszkek — azáltal, hogy másodlagos adattárolóként használjuk őket — megnövelhetik winchesterünk élettartamát. Ugyanis ahelyett, hogy telezsúfolnánk a merevlemezt a nem rendszeresen használt szoftverekkel, alkalmazásokkal és adatokkal, azokat — gyors betöltésre készen — inkább Floptical diszken tároljuk.

A Floptical diszkekhez természetesen külön meghajtóra van szükségünk. Az újonnan kifejlesztett drive nemcsak a Floptical diszkeket írja és olvassa, hanem a 720-as és a 1,44 MB-os, 3,5"-os floppykat is. A szabványos SCSI interfész kompatibilissá teszi a Floptical meghajtót minden ismert márkájú PC-vel.

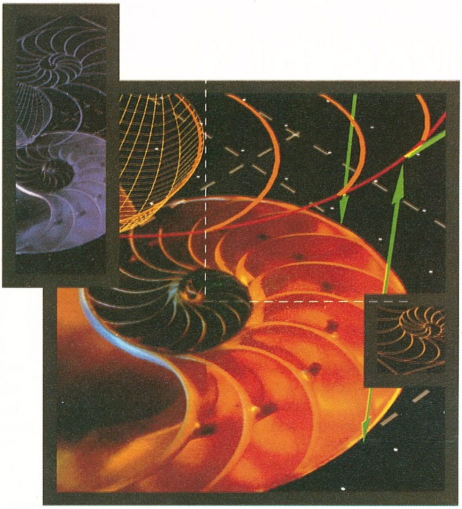
Mire ez a beharangozó napvilágot lát, már ismertek lesznek a termékek árai is, lapzártakor azonban csak az tűnik biztosnak, hogy a Floptical diszkkel a felhasználók könnyebben és gyorsabban cserélhetik ki, oszthatják meg egymással az adatállományaikat. Kíváncsian várjuk, hogy a megbízható, kompatibilis, szabványos, könnyen kezelhető — és a forgalmazó ígérete szerint költségtakarékos — új termék mennyire váltja be a hozzá fűzött reményeket!



Kedvezményes CAD-kabinet

Sokan a tavaszi Ifabo — s nem kevésbé a Camp '92 tapasztalatai alapján úgy gondolják, hogy elérkezett Magyarországra is a „CAD-time". A tavasszal végzett felmérések alapján a megkérdezett látogatók 30-35%-a CAD/CAM rendszerek iránt érdeklődött — konkrét vásárlási szándékkal. Valószínű, hogy ez a felismerés (is) ösztökélte a Sail-CAD Kft.-t arra, hogy egy CAD-oktatási kabinetet állítson fel a pár napja befejeződött Compfairen.

Az osztályteremben 6 pad (munkahely) áll az okulni vágyók rendelkezésére. A tanári asztalhoz kivetítő is tartozik.



Ha valaki ki szeretné próbálni valamelyik CAD programot, akkor ehhez kap egy rövid, 20-30 perces ismertetőt. Minden egyes szoftverhez készítenek olyan mintafeladatot, amelyet ez alatt a rövid idő alatt a delikvens — természetesen tanári segédlettel — meg is tud oldani. Az okítás végén a részvételt oklevéllel jutalmazzák a szervezők. Ez azonban nem szimbolikus diploma, hanem felmutatásával a „diákok" az év végéig kedvezményesen vásárolhatnak a Sail-CAD Kft. teljes hardver-szoftver kínálatából, amely a szabványos AutoDesk programok (AutoCAD, AutoSketch, AutoShade) mellett AutoCAD-alapú tervezőrendszerekből áll.

Ilyen például a BauCAD praxisorientált tervezőprogram, amely nem csupán térbeli építészeti szoftver: segítségével elvégezhető a vasbeton elemek szerkesztése, továbbá alkalmas nagyelemes (vázas, paneles) épületek tervezésére is. A szoftver generálja az ácsszerkezetet (fakivonatot, leszabási listát készít), valamint közművek is megszerkeszthetők a segítségével. A program az épülethez tartozó terepmodellt is elkészíti (belértve a kiemelendő földmennyiséget is). A forgalmazó állítása szerint Magyarországon nincs még egy ilyen komplex építészeti szoftver, mint a BauCAD. (Pedig igény az lenne!)

Egyelőre még csak németül beszél a BauCAD, de tervezik a szoftver magyarítását. Ugyanis a CAMP-en jelent meg az AutoCAD Release 12, amelyet máris magyarít az AutoDesk Iroda. S így az AutoCAD új verziójával a BauCAD legújabb változata magyar menükkel, hibaüzenetekkel és kézikönyvvel áll majd a felhasználók rendelkezésére. Az AutoCAD-alapú szoftver tudásához képest nem drága, 332 000 Ft (+ a magyar AutoCAD Release 12: 225 000 Ft). A forgalmazó hardverrel együtt is értékesíti a szoftvert. A komplett munkahely (egy megfelelő Elonex PC, egy 17"-os monitor és egy tintasugaras A/2-es nyomtató) ára a szoftverrel együtt kb. másfél millió forint. Az így összeállított munkahely ára kevesebb lesz, mintha darabonként vennénk meg a hardvert és a szoftvert. S az oktatási kabinetben szerzett diploma feljogosít minden résztvevőt a kedvezményes árra is. Aki ilyen jellegű beruházás előtt áll, éljen a felkínált lehetőséggel!

Csőd a csődben

Egyes hírek szerint ma Magyarországon az összesen rendelkezésre álló 8, azaz nyolc bíróra mintegy 5000 felszámolási eljárás jogi rendezése vár. Évekig húzódhat tehát egy-egy cég agóniája, s előbb-utóbb a hitelezők is kénytelenek csődöt jelenteni. A fennmaradni kívánó cégeknek pedig elemi érdekük, hogy naprakész információkkal rendelkezzenek a csődbe ment vagy felszámolandó vállalatokról. Ezek a létfontosságú információk most már nemcsak a Heti Csődértesítő című újság alapján szerezhetők be (hosszas manuális hasonlítgatásokkal), hanem a CompuDrug Standard Kft. által kifejlesztett számítógépes csődfigyelési rendszer segítségével is.

A szoftverfejlesztőknek a legtöbb fejfájást a pontatlan cégmegnevezések okozták, ugyanis sokan a hosszú cégnevek helyett egyéni rövidítéseket használnak. A probléma megoldására — egyúttal a hibák kiküszöbölésére — belső azonosítókkal látták el valamennyi, a Cégközlönyben szereplő vállalatot. További újításuk, hogy az adatbázist csak szaklektorok bővíthetik. Ez drágább ugyan, és lassúbb bevitelt jelent, de így egy cég garantáltan csak egyszer szerepel majd az adatbázisban, hiszen név, cím, irányítószám, utca... alapján vizsgálják az azonosságokat.

Ha valamely vállalat „gyanús", akkor valóságos kutatómunkába kezdenek. Nemcsak a múlt év végéig megszűnt vállalatok katalógusában keresgélnek, hanem elmennek akár a Cégbíróságra is. S ha még mindig nem látnak tisztán, akkor a megrendelőhöz fordulnak a meglévő adatok pontosítása érdekében.

A csődfigyelő programmal elsősorban szolgáltatni kíván a CompuDrug Standard Kft. A megrendelő átadja (papíron vagy floppyn) a vele kapcsolatban álló vállalatok listáját. Hetente egyszer futtatják a programot. A megrendelőt levélben, faxon vagy táviratban értesítik a csődbe ment vagy felszámolásra került partnervállalatokról. A rendszer figyeli a felszámolási időintervallumokat is. A CompuDrug Standard Kft. kívánság szerint figyeli a meghirdetett csődtárgyalások időpontjait, helyét, ezekről akár naponta is értesíti a megrendelőket.

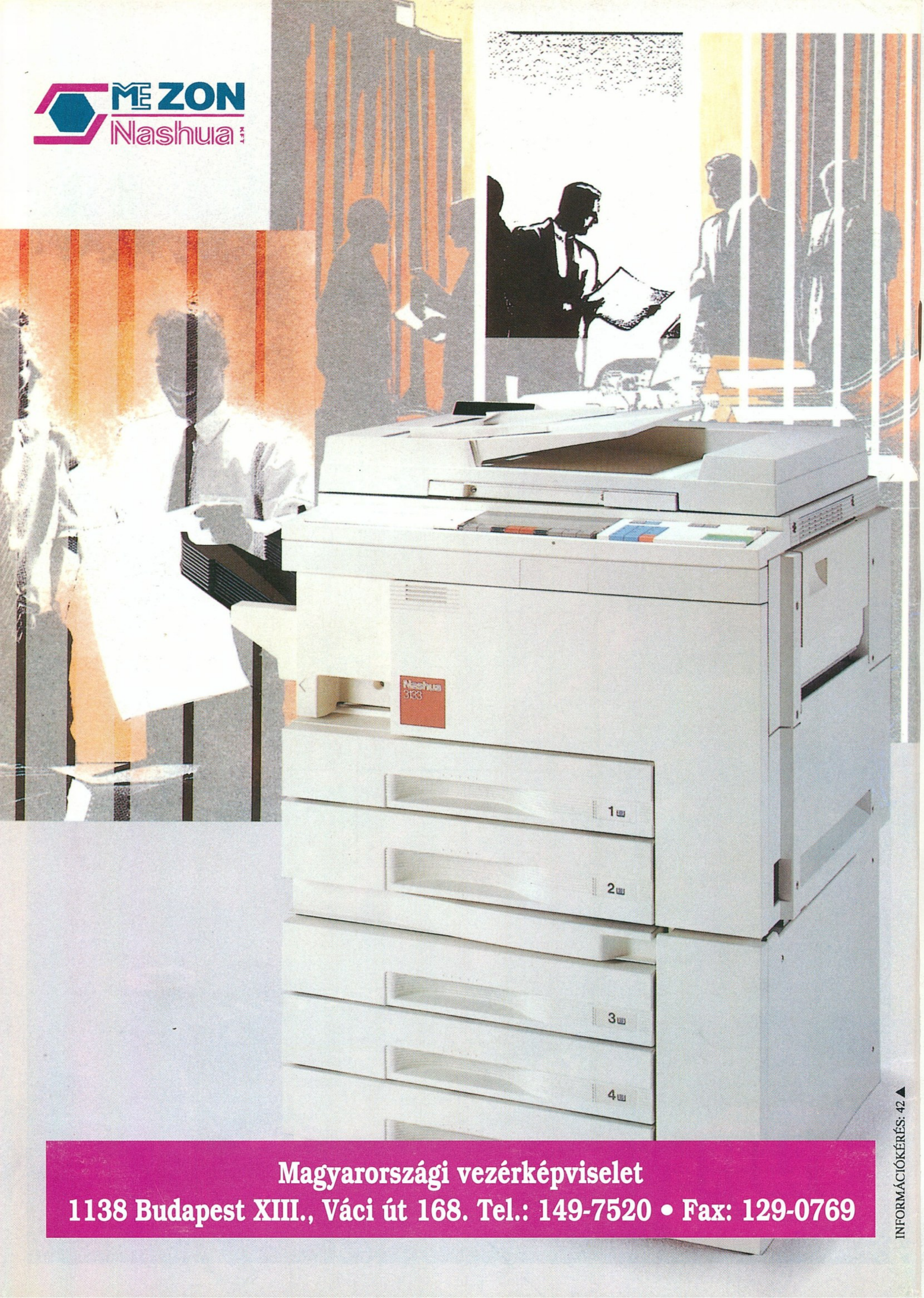
Speciális szolgáltatásuk közé tartozik, hogy „megkülönböztetett figyelmet" szentelnek a már csődbe ment cégek csődbe ment partnercégeinek. Az ebből származó információkkal egyrészt hatalmas pénzeket lehet megfogni, másrészt a perspektivikus gondolkodást is könnyítik.

Természetesen a csődfigyelés nemcsak szolgáltatásként vehető igénybe, hanem maga a szoftver és az adatbázis is megvásárolható. A kb. 50 000 forintos program és 80 000 forint körüli adatbázis — méretfüggően — igazán méltányos ár ahhoz képest, hogy mennyit takaríthatunk meg használatával.

A rendszert Clipperben fejlesztették, a keresések gyorsítására Turbo Pascal rutinokat írtak. A rendkívüli szakmai alapossággal készült szoftver a Cégbíróságon bejegyzett cégeket figyeli, így az egyéni vállalkozók, alapítványok nem szerepelnek az adatbázisban. A programnak azonban van egy szépséghibája: nem ismeri a magyar ékezetes karaktereket. Első körben ez igen meglepő, helyenként értelemzavaró. A fejlesztők állítása szerint ezzel az egyszerűsítéssel a hibaszázalék jelentősen csökkenthető, és a keresések lényegesen gyorsíthatók. Ez így igaz, de manapság már egy magyar felhasználó jogos elvárása, hogy ékes magyar nyelven olvashassa a neki szóló információkat.

Sziebig Andrea


EGY DÖNTÉSHEZ NÉHA ELÉG EGYETLEN ÉRV...
Mi a Fuji floppyk mellett ennél sokkal több érvvel szolgálhatunk.
Úgy körülbelül harmincmillióval.
Ennyiszer futtatható le egy Fuji floppy számottevő minőségromlás nélkül.
Ez tisztes munkaidővel, ötnapos munkahéttel és 10 másodperces leolvasási idővel számolva 80 éves feladatot jelentene - ha valaki éppen nem találna jobb elfoglaltságot.
E mögött a fantasztikus szám mögött azonban nem boszorkányság, hanem hosszú évek alapos fejlesztőmunkája áll.
Floppyjaink védőburka különleges, hőálló műanyagból készül, mely hirtelen hőmérsékletváltozás esetén sem vetemedik meg, így nem zavarja a leolvasást.
A lemez adathordozó mágneses részecskéi nem egy irányba rendezetten, hanem véletlenszerű eloszlásban állnak, így az adatfelvitel biztosabb.
A lemezek újszerű tisztító-mechanikája pedig garantálja, hogy a Fuji floppykkal nem kerül porszem a gépezetbe...
FUJIFILM MAGYARORSZÁG KFT.
1088 Budapest, Rákóczi u. 1-3.
Tel.: 266-6218, 266-4563,
267-6944, 117-7770/347, 348
Fax: 266-2742
FUJI FILM
MD2D
Super HR
COLOR DISKS
FUJI
A CSÚCS, AMELY MEGHÓDÍTOTTA AZ EMBERT.
DDB NEEDHAM


MEZON
Nashua
Nashua 3133
Magyarországi vezérképviselet
1138 Budapest XIII., Váci út 168. Tel.: 149-7520 • Fax: 129-0769
INFORMÁCIÓKÉRÉS: 42 ▲